

JAVA 코드

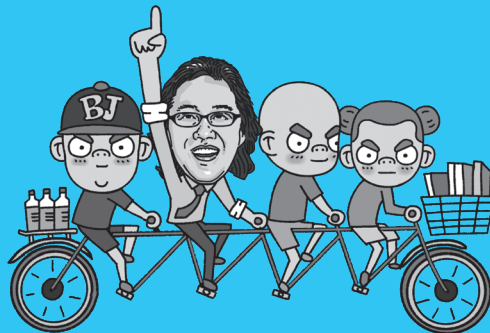
3장 기본 알고리즘 - 수열

4장 기본 알고리즘 - 수학

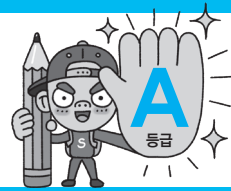
5장 응용 알고리즘 - 자료 구조

6장 응용 알고리즘 - 배열

7장 실무 응용 알고리즘



010 수열 $1 - 1+2+3+4+ \dots +100$ 까지의 합계



상기 17.4

문제

1+2+3+4+...+100까지의 합계를 구하는 순서도를 작성하십시오.

JAVA 코드

```

public class Section010 {
    public static void main(String[] args) {
        ❶ int i, j; 정수형 변수 i와 j를 선언한다.
        ❷ i = 0;
        ❷ j = 0;
        do do~while 반복문의 시작점으로 ❸~❹번 문장을 반복하여 수행한다.
        {
            ❸ i++; 'i = i + 1;'과 동일하다. i의 값을 1씩 누적시킨다.
            ❹ j += i; 'j = j + i;'와 동일하다. i의 값을 j에 누적시킨다.
            ❺ } while (i < 100); do~while 반복문의 조건이다. i가 100보다 작은 동안 ❸~❹번 문
                장을 반복하여 수행한다.
            ❻ System.out.printf("%d %d", i, j); 서식 문자열 "%d"에 대응하는 정수 변수
                i와 j의 값 100과 5050을 출력한다.
        }
    }
}

```



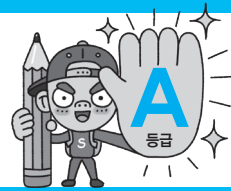
홀수의 합계 계산하기

문제 1+3+5+7+...+99의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```
public class Section010_Wait {
    public static void main(String[] args) {
        ①      int i, j;   정수형 변수 i와 j를 선언한다.
        ②      i = -1;
        ②      j = 0;
        do   do~while 반복문의 시작점으로 ③~④번 문장을 반복하여 수행한다.
        {
            ③      i += 2;   'i = i + 2'와 동일하다. i의 값을 2씩 누적시킨다.
            ④      j += i;   'j = j + i'와 동일하다. i의 값을 j에 누적시킨다.
            ⑤      } while (i < 99);   do~while 반복문의 조건이다. i가 99보다 작은 동안 ③~④
                                   번 문장을 반복하여 수행한다.
        ⑥      System.out.printf("%d", j);      결과 2500
    }
}
```

011

수열 $2 - 1 - 2 + 3 - 4 + 5 - 6 + \dots + 99 - 100$ 의 합계

유형

1

스위치 변수 이용하기

상기 09.7

문제

1-2+3-4+5-6...+99-100까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```

public class Section011_Type1 {
    public static void main(String[] args) {
        ① int i, j, sw;
        ② i = j = 0; 'i = 0; j = 0;'과 동일하다. i와 j를 각각 0으로 초기화한다.
        ③ sw = 0;
        do do~while 반복문의 시작점으로 ④~⑨번 문장을 반복하여 수행한다.
        {
            ④ i++;
            ⑤ if (sw == 0) sw가 0이면 ⑥~⑦번 문장을 실행하고, 아니면 ⑧~⑨번 문장으로 이동해
                서 실행을 계속한다.
            {
                ⑥ j += i; ⑥번의 조건식이 참일 경우 실행할 문장이다. 'j = j + i;'와 동일하다. i의
                    값을 j에 누적시킨다.
                ⑦ sw = 1; 실행 후 제어가 ⑩번으로 이동한다.
            }
            else ⑤번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            {
                ⑧ j -= i; ⑧번의 조건식이 거짓일 경우 실행할 실제 처리문이다. 'j = j - i;'와 동
                    일하다. - i의 값을 j에 누적시킨다.
                ⑨ sw = 0; 실행 후 제어가 ⑩번으로 이동한다.
            }
        } while (i < 100); do~while 반복문의 조건이다. i가 100보다 작은 동안 ④~⑨번
        문장을 반복하여 수행한다.
        ⑪ System.out.printf("%d", j); 결과 -50
    }
}

```

유형

2

순서에 의한 반복 계산

문제

1-2+3-4+5-6+...-98+99까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```

public class Section011_Type2 {
    public static void main(String[] args) {
        ❶ int i = 0, j = 0;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 ❷~❸번 문장을 무한 반복한다.
        {
            ❷ i++;
            ❸ j += i;
            ❹ if (i >= 99) if문으로 i가 99 보다 크거나 같으면 ❷번 문장을 실행하고, 아니면 ❺
                ~❻번 문장으로 이동해서 실행을 계속한다.
            {
                ❷ System.out.printf("%d", j); 결과: 50
                break; 반복문(while)을 탈출한다. 수행 후 프로그램을 종료한다.
            }
            else ❹번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            {
                ❺ i++;
                ❻ j -= i; 수행 후 제어는 while문의 시작점으로 이동하여 ❷번부터 다시 시작한다.
            }
        } while 반복문의 끝이다.
    }
}

```

문제

$(-1) \times 2 \times (-3) \times 4 \times (-5) \times \dots \times 100$ 의 합계를 구하는 순서도를 작성하시오.

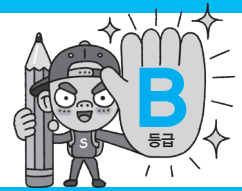
JAVA 코드

```

public class Section011_Type3 {
    public static void main(String[] args) {
        ❶ int i;
        ❶ double j; 실수형 변수 j를 선언한다.
        ❷ i = 0;
        ❸ j = 1;
        do do~while 반복문의 시작점으로 ❹~❷번 문장을 반복하여 수행한다.
        {
            ❹ i++;
            ❺ if (i % 2 == 0) i를 2로 나눈 나머지가 0이면 ❻번 문장을 실행하고, 아니면 ❷
                ❷ 번 문장으로 이동해서 실행을 계속한다.
            ❻ j *= i; 'j = j * i;'와 동일하다. j * i의 값을 j에 저장시킨다. 수행 후
                제어가 ❸번으로 이동한다.
            else ❺번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            ❷ j *= i * -1; 'j = j * (i * (-1))'과 동일하다. 수행 후 제어가 ❸번으로 이동한다.
            ❸ } while (i < 100); i가 100보다 작은 동안 ❹~❷번 문장을 반복하여 수행한다.
            ❹ System.out.printf("%11.4e", j); 서식 문자열 "%11.4e"에 대응하는 실수 변수
                의 값 9.3362154×1097를 소수점 이상 11자리, 소수점 이하 4자리를 잡아 9.3326e+157을 출력한다.
        }
    }
}

```

012 수열 $3 - -(\frac{1}{2}) + (\frac{2}{3}) - (\frac{3}{4}) + (\frac{4}{5}) - (\frac{5}{6}) \cdots - (\frac{99}{100})$ 의 합계



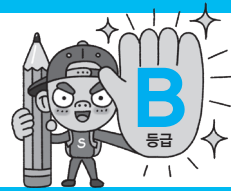
문제

$-(\frac{1}{2}) + (\frac{2}{3}) - (\frac{3}{4}) + (\frac{4}{5}) - (\frac{5}{6}) \cdots - (\frac{99}{100})$ 의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```
public class Section012 {
    public static void main(String[] args) {
        ① float i = 0, j = 0; 실수형 변수 i와 j를 선언한다.
        do do~while 반복문의 시작점으로 ②~⑥번 문장을 반복하여 수행한다.
        {
            ② i++;
            ③ if ((int)(i / 2) == i / 2) (int)(i / 2)와 i / 2의 값이 같으면 ④번 문장
                을 실행하고, 아니면 ⑤번 문장으로 이동해서
                실행을 계속한다.
            ④ j += i / (i + 1); 수행 후 제어기가 ⑥번으로 이동한다.
            else ③번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            ⑤ j -= i / (i + 1); 수행 후 제어기가 ⑥번으로 이동한다.
            ⑥ } while (i < 99); i가 99보다 작은 동안 ②~⑤번 문장을 반복하여 수행한다.
            ⑦ System.out.printf("%f", j); 결과 -0.688172
                결과는 실수로 반환되기 때문에 서식 문자열 '%f'
                를 사용한다.
        }
    }
}
```

013 수열4 - 1+2+4+7+11+16+22+...로 증가하는 수열



유형

1

항 사이의 증가하는 값이 일정한 비율로 증가하는 수열

문제

1+2+4+7+11+16+22+...의 순서로 나열되는 수열의 20번째 항까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```

public class Section013_Type1 {
    public static void main(String[] args) {
        ❶ int i, j, k;
        ❷ i = 0;
        ❸ j = 1;
        ❹ k = 1;
        do do~while 반복문의 시작점으로 ❹~❷번 문장을 반복하여 수행한다.
        {
            ❺ i++;
            ❻ j += i;
            ❼ k += j;
        } while (i < 19); i가 19보다 작은 동안 ❹~❷번 문장을 반복하여 수행한다.
        ❾ System.out.printf("%d", k); 결과 1350
    }
}

```



문제

1+3+6+10+15+21+28+...의 순서로 나열되는 수열의 20번째 항까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```
public class Section013_Wait {
    public static void main(String[] args) {
        ❶ int i = 0, j = 0, k = 0;
          do do~while 반복문의 시작점으로 ❷~❹번 문장을 반복하여 수행한다.
          {
            ❷ i++;
            ❸ j += i;
            ❹ k += j;
            ❺ } while (i < 20); i가 20보다 작은 동안 ❷~❹번 문장을 반복하여 수행한다.
            ❻ System.out.printf("%d", k);      결과 1540
          }
    }
}
```

문제

$-1+2-4+7-11+16-22+\dots$ 의 순서로 나열되는 수열의 20번째 항까지의 합계를 구하십시오.

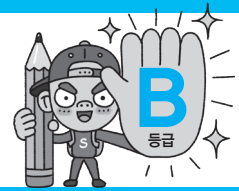
JAVA 코드

```

public class Section013_Type2 {
    public static void main(String[] args) {
        ① int i, j, k, L;
        ② i = 0;
        ③ j = 1;
        ④ k = -1;
        ⑤ L = -1;
        do do~while 반복문의 시작점으로 ⑥~⑨번 문장을 반복하여 수행한다.
        {
            ⑥ i++;
            ⑦ j += i;
            ⑧ L *= -1; 'L = L * (-1)'과 동일하다. L * -1의 값을 L에 저장시킨다.
            ⑨ k += j * L;
        ⑩ } while (i < 19); i가 19보다 작은 동안 ⑥~⑨번 문장을 반복하여 수행한다.
        ⑪ System.out.printf("%d", k); 결과 100
    }
}

```

014 수열5 - $1!+2!+3!+4!+5!+\dots+10!$ 의 합계



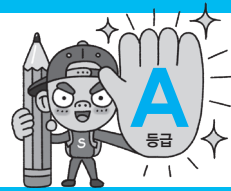
문제

$1!+2!+3!+4!+5!+\dots+10!$ 의 순서로 나열되는 수열의 10번째 항까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```
public class Section014 {
    public static void main(String[] args) {
        ❶ int i = 1, k = 1, j = 1;
          do~while 반복문의 시작점으로 ❷~❹번 문장을 반복하여 수행한다.
          {
            ❷ i++; i의 값을 1씩 누적시킨다.
            ❸ j *= i; j * i의 값을 j에 저장시킨다.
            ❹ k += j; j의 값을 k에 누적시킨다.
            ❺ } while (i < 10); i가 10보다 작은 동안 ❷~❹번 문장을 반복하여 수행한다.
            ❻ System.out.printf("%d", k); 결과 4037913
          }
    }
}
```

015 수열6 - 피보나치 수열(1+1+2+3+5+8+13+...)의 합계



기사 18.10, 05.10

문제

1+1+2+3+5+8+13+...의 순서로 나열되는 피보나치 수열의 20번째 항까지의 합계를 구하는 순서도를 작성하시오.

JAVA 코드

```
public class Section015 {
    public static void main(String[] args) {
        ① int hap, cnt, c;
        ① int a = 1, b = 1;
        ② hap = 2;
        ③ cnt = 2;
        while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
                       지 { } 사이의 문장을 무한 반복한다.
        {
            ④ c = a + b;
            ⑤ hap += c;  c의 값을 hap에 누적시킨다.
            ⑥ cnt++;  cnt의 값을 1씩 누적시킨다.
            ⑦ if (cnt < 20)  cnt가 20보다 작으면 ⑧~⑩번 문장을 실행하고, 아니면 ⑩번 문장으
                       로 이동해서 실행을 계속한다.
            {
                ⑧ a = b;
                ⑨ b = c;  수행 후 제어기가 while문의 시작점으로 이동하여 ④번부터 다시 시작한다.
            }
            else  ⑦번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            {
                ⑩ System.out.printf("%d", hap);  결과 17710
                break;  반복문(while)을 탈출하여 프로그램을 종료한다.
            }
        }  while 반복문의 끝이다.
    }
}
```



JAVA 코드

```

import java.util.Scanner;  Scanner() 메소드가 정의되어 있는 클래스이다.

public class Section015_Wait {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);  Scanner 클래스
                                                    의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.
                                                    System.in은 표준 입력장치, 즉 키보드를 의미한다.

❶      int a, b, y, n, k, c;
❷      a = 1;
❸      b = 1;
❹      y = 2;
❺      n = scanf.nextInt();  키보드로부터 정수형 값을 입력받아 n에 저장한다.
❻      for (k = 3; k <= n; k++)  반복 변수 k가 3에서 시작하여 1씩 증가(증가값)하
                                   면서 n보다 작거나 같은 동안 ❸~❹번 문장을 n
                                   회 반복한다.

        {
❻          c = a + b;
❼          y += c;
❸          a = b;
❹          b = c;
        }
❺      System.out.printf("%d", y);
        scanf.close();  Scanner 클래스의 객체 변수는 임의의 메모리 영역을 확보하여 사
                           용하는 것이므로 프로그램 종료 전에 사용 하던 메모리 영역을 해제
                           해야 다른 프로그램이 해당 영역을 사용할 수 있다.

    }
}

```



기사 18년 10월

문제 4 다음은 피보나치 수열의 합계를 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Problem {  
    public static void main(String[ ] args) {  
        ①      int a, b, c, sum      정수형 변수 a, b, c, sum을 선언한다.  
        ②      a = b = 1;           a와 b 변수에 1을 저장한다.  
        ③      sum = a + b;         a와 b를 더한 값을 sum에 저장한다.  
  
        ④      for(int i = 3; i<=5; i++)      정수형 변수 i를 반복 변수로 선언하면서 초기값으로 3을 갖고, 1씩 증가하면서 5보다 작거나 같은 동안 ⑤~⑧ 사이의 문장을 반복하여 수행한다. 결국 ⑤~⑧번 문장을 총 3회 반복한다.  
  
        {  
            ⑤      c = a + b;         a와 b를 더한 값을 c에 저장한다.  
            ⑥      sum += c;         'sum = sum + c'와 동일하다. c의 값을 sum에 누적한다.  
            ⑦      a = b;           b의 값을 a에 저장한다.  
            ⑧      b = c;           c의 값을 b에 저장한다.  
        }  
        ⑨      System.out.println(sum);      sum의 값을 출력한 후 커서를 다음 줄 처음으로 옮긴다.  
    }  
}
```



문제 5 다음은 $(77 \times 1) + (76 \times 2) + (75 \times 3) + \dots + (2 \times 76) + (1 \times 77)$ 의 결과를 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test05 {  
    public static void main(String[] args) {  
        ❶      int q, m;   변수 q, m을 선언한다.  
                int p = 0, h = 0;   p는 1씩 누적되며 증가하는 변수이므로 p를 0으로 초기화한다.  
                                    h는 각 항의 값이 계산되어 저장되므로 0으로 초기화한다.  
        do   do~while 반복문의 시작점으로 ❷~❺번을 반복 수행한다.  
        {  
            ❷      p++;           p를 1부터 77까지 증가시키기 위해 p를 1씩 증가시킨다.  
            ❸      q = 78 - p;     q를 77부터 1까지 감소시키기 위해 q에 78-p 값을 저장한다.  
            ❹      m = q * p;     수열의 각 항을 만든다.  
            ❺      h += m;        수열의 항 m을 합계 h에 누적한다.  
            ❻      } while (p < 77);   p가 77보다 작은 동안 ❷~❺번 문장을 반복 수행한다.  
            ❼      System.out.printf("%d", h);   h를 출력하고 끝낸다.  
        }  
    }  
}
```



문제 6 다음은 $1+1/1+1/2+1/3+1/4+\dots+1/10$ 까지의 합을 구하여 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test06 {  
    public static void main(String[] args) {  
        ① float hap, a, d; 실수형 변수 hap, a, d를 선언한다.  
        ② hap = a = 1; 처음 더해지는 값 1을 가지고 시작하기 위해 hap을 1로 초기화한다.  
                처음 분모 값이 1이므로 a를 1로 초기화한다.  
        do do~while 반복문의 시작점으로 ③~⑤번을 반복 수행한다.  
        {  
            ③ d = 1 / a; 각 항의 분수(1/1, 1/2, ...)를 만들어 d에 저장한다.  
            ④ hap += d; 합계 hap에 d를 누적한다.  
            ⑤ a++; 분모를 증가시킨다.  
        } while (a <= 10); a가 10보다 작거나 같은 동안 ③~⑤번을 반복 수행한다.  
        ⑦ System.out.printf("%f", hap); hap을 출력하고 끝낸다.  
    }  
}
```

016 소수 판별



유형

1

나누어 떨어지지 않을 때

상기 20, 11, 18, 6

문제

1보다 큰 임의의 정수를 입력하여 소수를 판별하는 순서도를 작성하시오.

JAVA 코드

```

import java.util.Scanner; Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section016_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in); Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int a, i, j;
        ② a = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 a에 저장한다.
        ③ i = a - 1;
        ④ j = 2;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 { } 사이의 문장을 무한 반복한다.
        {
            ⑤ if (j <= i) j가 i보다 작거나 같으면 ⑦번 문장을 실행하고, 아니면 ⑥번 문장으로
                이동해서 실행을 계속한다.
            {
                ⑦ if (a % j == 0) a를 j로 나눈 나머지가 0이면 ⑧번 문장을 실행하고, 아니
                    면 ⑨번 문장으로 이동해서 실행을 계속한다.
                {
                    ⑧ System.out.printf("소수 아님");
                        break; 반복문(while)을 탈출한다. 제어가 ⑩번으로 이동한다.
                }
                ⑨ else ⑦번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
                    j++; j의 값을 1씩 누적시킨다. 수행 후 제어가 while문의 시작점으로 이동
                    하여 ⑤번부터 다시 시작한다.
            }
            else ⑤번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            {
                ⑥ System.out.printf("소수");
                    break; 반복문(while)을 탈출한다. 제어가 ⑩번으로 이동한다.
            }
        }
        ⑩ scanf.close(); 객체 변수 scanf의 메모리 영역을 해제한다.
    }
}

```

문제

1보다 큰 임의의 정수를 입력하여 소수를 판별하는 순서도를 작성하시오.

JAVA 코드

```

import java.util.Scanner;  Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section016_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);  Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int a, j;
        ② a = scanf.nextInt();  키보드로부터 정수형 값을 입력받아 a에 저장한다.
        ③ j = 2;
        ④ while (a % j != 0)  a를 j로 나눈 나머지가 0이 아니면 ⑤번 문장을 반복하여 수행한다.
        ⑤     j++;
        ⑥     if (a == j)  a와 j가 같으면 ⑦번 문장을 실행하고, 아니면 ⑧번 문장으로 이동해서 실행을
                        계속한다.
        ⑦         System.out.printf("소수");
        else  ⑥번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
        ⑧         System.out.printf("소수 아님");
        scanf.close();  객체 변수 scanf의 메모리 영역을 해제한다.
    }
}

```

유형

3

제공된 이용하기

산기 20.11

문제

1보다 큰 임의의 정수를 입력 받아 소수인지 판별하는 순서도를 작성하시오.
소수이면 “소수”를, 소수가 아니면 “소수 아님”을 출력하시오.

JAVA 코드

```
import java.util.Scanner; Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section016_Type3 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in); Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int a, j;
        ② a = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 a에 저장한다.
        ③ j = 2;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 { } 사이의 문장을 무한 반복한다.
        {
            ④ if (j <= Math.sqrt(a)) j가 a의 제곱근보다 작거나 같으면 ⑤번 문장을 실행하고, 아니면 ⑦번 문장으로 이동해서 실행을 계속
                한다. C언어에서 math.h에 들어있던 함수들은 대개 Math 클래스에서 찾아볼 수 있다.
            {
                ⑤ if (a % j == 0) a를 j로 나눈 나머지가 0이면 ⑥번 문장을 실행하고, 아니면 ⑧번 문장을 실행한다.
                {
                    ⑥ System.out.printf("소수 아님");
                    break; 반복문(while)을 탈출한다. 제어가 ⑨번으로 이동한다.
                }
                else ⑤번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
                {
                    ⑧ j++;
                }
            }
            else ④번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
            {
                ⑦ System.out.printf("소수");
                break; 반복문(while)을 탈출한다. 제어가 ⑨번으로 이동한다.
            }
        }
        ⑨ scanf.close(); 객체 변수 scanf의 메모리 영역을 해제한다.
    }
}
```

017 소수의 합 구하기



문제

임의의 정수를 입력 받아 그 안에 포함된 소수의 합을 구하는 순서도를 작성하시오. 예를 들어 10을 입력 받았다면 2~10 사이에 속한 소수의 합을 계산한다.

JAVA 코드

```
import java.util.Scanner; Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section017 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in); Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int a, hap, k, j;
        ② a = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 a에 저장한다.
        ③ hap = 0;
        ④ k = 2;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ⑤ j = 2;
            ⑥ while (k % j != 0) k를 j로 나눈 나머지가 0이 아니면 ⑦번 문장을 반복 수행한다.
            ⑦ j++;
            ⑧ if (k == j) k와 j가 같으면 ⑨번 문장을 실행하고, 아니면 ⑩번 문장으로 이동해서 실행을 계속한다.
            ⑨ hap += k; hap의 값에 k를 누적시킨다. 수행 후 제어기가 ⑩번으로 이동한다.
            ⑩ if (k < a) k가 a보다 작으면 ⑪번 문장을 실행하고, 아니면 ⑫번 문장으로 이동해서 실행을 계속한다.
            ⑪ k++; 수행 후 제어기가 while문의 시작점으로 이동하여 ⑤번부터 다시 시작한다.
            else
            {
                ⑫ System.out.printf("%d", hap);
                break; 반복문(while)을 탈출한다. 제어기가 ⑬번으로 이동한다.
            }
        }
        ⑬ scanf.close(); 객체 변수 scanf의 메모리 영역을 해제한다.
    }
}
```

018 소수의 개수 구하기



문제

배열 A[99]에 2~100 사이의 정수를 기억시킨 후 이 배열을 이용하여 소수의 개수를 구하시오.

JAVA 코드

```

public class Section018 {
    public static void main(String[] args) {
        ① int k, i, j, m;
        ① int a[] = new int[99];
        ② k = 1;
        do ③~while 반복문의 시작점으로 ③, ④번 문장을 반복하여 수행한다.
        {
            ③ k++;
            ④ a[k - 2] = k; 배열의 위치를 0부터 시작하므로 배열 a의 첫 번째 위치인 A[0]에 k를 저장하기 위해 A[k-2]로 지정한다.
            ⑤ } while (k < 100);
            ⑥ i = -1; 배열 a의 첨자 i는 'i++(⑦)'을 수행한 후 0이 되도록 -1로 초기화한다.
            ⑥ j = 0;
            while (true) ⑦~while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ⑦~⑮번 문장을 무한 반복한다.
            {
                ⑦ i++;
                ⑧ if (i > 98) 배열의 위치를 0부터 시작하기 때문에 행 첨자 i는 0에서 98까지 99회 반복하여 수행한다.
                    i가 98보다 크면 ⑨번 문장을 실행하고, 아니면 ⑩번 문장으로 이동해서 실행을 계속한다.
                {
                    ⑨ System.out.printf("%d", j);
                    break; 반복문(while)을 탈출한다. 제어가 ⑮번으로 이동한다.
                }
                ⑩ if (a[i] == 0) a[i]가 0이면 continue에 의해 ⑪~⑮번 문장을 생략하고 while 반복문의 시작점으로 이동해서 실행하고,
                    아니면 ⑪번 문장으로 이동해서 실행을 계속한다.
                    continue; 이후의 문장, 즉 ⑪~⑮번 문장을 생략하고 while 반복문의 시작점으로 이동한다.
                ⑪ j++;
                ⑫ m = i;
                while (true) ⑬~while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ⑬~⑮번 문장을 무한 반복한다.
                {
                    ⑬ m += a[i]; 'm = m + a[i]'와 동일하다.
                    ⑭ if (m > 99) m이 99보다 크면 다음 문장을 실행하고, 아니면 ⑮번 문장으로 이동해서 실행을 계속한다.
                        break; 반복문(while)을 탈출한다. 수행 후 제어가 바깥쪽 while문의 시작점, 즉 ⑦번 문장으로 이동한 후 ⑦번부터 다시 시작한다.
                    ⑮ a[m] = 0; 수행 후 제어가 안쪽 while문의 시작점, 즉 ⑬번 문장으로 이동한 후 ⑬번부터 다시 시작한다.
                }
            }
        }
    }
}

```

019 최대공약수, 최소공배수



기사 16.4, 10.4

문제

두 수를 입력 받아 두 수의 최대공약수와 최소공배수를 계산해서 출력하는 순서도를 작성하시오.

JAVA 코드

```
import java.util.Scanner; Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section019 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in); Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int a, b, big, small, mok, nmng, gcm, lcm;
        ② a = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 a에 저장한다.
        ② b = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 b에 저장한다.
        ③ if (a >= b) a가 b보다 크거나 같으면 ④번 문장을 실행하고, 아니면 ⑥번 문장으로 이동해서 실행을 계속한다.
        {
            ④ big = a;
            ⑤ small = b; 수행 후 제어기가 ⑦번 아래쪽으로 이동한다.
        }
        else ③번의 조건식이 거짓일 경우 실행할 문장의 시작점이다.
        {
            ⑥ big = b;
            ⑦ small = a; 수행 후 제어기가 ⑦번 아래쪽으로 이동한다.
        }
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ⑧ mok = big / small;
            ⑨ nmng = big - mok * small;
            ⑩ if (nmng == 0) nmng가 0이면 ⑩번 문장을 실행하고, 아니면 ⑪번 문장으로 이동해서 실행을 계속한다.
            {
                ⑬ gcm = small;
                ⑭ lcm = a * b / gcm;
                ⑮ System.out.printf("%d %d", gcm, lcm);
                break; while 반복문을 탈출한다. 제어기가 ⑮번으로 이동한다.
            }
            ⑪ big = small;
            ⑫ small = nmng; 수행 후 제어기가 while문의 시작점으로 이동하여 ③번부터 다시 시작한다.
        }
        ⑯ scanf.close();
    }
}
```

020 약수 구하기



기사 18.6, 13.7, 11.4, 07.4

문제

정수를 입력 받아 약수를 구해 출력하는 순서도를 작성하시오.

JAVA 코드

```
import java.util.Scanner; Scanner() 함수가 정의되어 있는 헤더 파일이다.

public class Section020 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in); Scanner 클래스의 객체 변수 scanf를 키보드로 입력받을 수 있도록 생성한다.

        ① int b, mok, nmg, i;
        ① int a[] = new int[100];
        ② b = scanf.nextInt(); 키보드로부터 정수형 값을 입력받아 b에 저장한다.
        ③ int c = 0, d = -1; 배열의 위치가 0부터 시작하므로 배열 a의 첨자 d는 'd++(⑩)'을 수행한 후 0이 되도록 -1로 초기화한다.
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ④ c++;
            ⑤ if (c <= b) c가 b보다 작거나 같으면 ⑨번 문장을 실행하고, 아니면 ⑥번 문장으로 이동하여 실행을 계속한다.
            {
                ⑨ mok = b / c;
                ⑩ nmg = b - mok * c;
                ⑪ if (nmg == 0)
                {
                    ⑫ d++;
                    ⑬ a[d] = c; 수행 후 제어기가 while문의 시작점으로 이동하여 ④번부터 다시 시작한다.
                }
            }
            else
            {
                ⑥ System.out.printf("%d ", b);
                ⑦ for (i = 0; i <= d; i++) 배열의 위치가 0부터 시작하기 때문에 첨자 i는 0부터 d가 될 때까지 ③번 문장을 반복 수행한다.
                ⑧ System.out.printf("%d ", a[i]);
                break; 반복문(while)을 탈출한다. 제어기가 ⑭번으로 이동한다.
            }
        }
        ⑭ scanf.close(); 객체 변수 scanf의 메모리 영역을 해제한다.
    }
}
```

021 소인수 분해하기



선기 20.5, 12.10, 06.11, 기사 19.10

문제

정수를 입력 받아 소인수를 구해 출력하는 순서도를 작성하시오.

JAVA 코드

```

import java.util.Scanner;

public class Section021 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int b, c, d, e, mok = 0, nmg;
        ❷ int a[] = new int[100];
        ❸ b = scanf.nextInt();
        ❹ c = -1;
        ❺ while (true)
        {
            d = 2;
            e = (int)Math.sqrt(b);
            while (true)
            {
                ❻ if (d > e)
                {
                    ❼ d = b;
                    break;
                }
                mok = b / d;
                nmg = b - mok * d;
                ❽ if (nmg == 0)
                {
                    break;
                }
            }
        }
    }
}

```

JAVA에서는 변수의 값이 정의되지 않고 사용되거나 사용될 가능성이 있는 경우 오류가 발생한다. ❸번에서 d가 e보다 큰 경우 ❹번을 수행하지 않고 ❺번으로 진행되었을 때 정의되지 않은 알 수 없는 값이 mok에 저장되어 있어 실행시 오류가 발생하므로, C언어와 다르게 mok을 0으로 초기화하였다.

❸번 문장을 무한 반복한다. 조건이 참(true)이므로 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

❹번 문장을 무한 반복한다. 조건이 참(true)이므로 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

입력 받은 값 b의 제곱근을 구해 정수로 변환한 후 e에 저장한다. b가 5라면 5의 제곱근은 2.236...이지만 (int)에 의해 정수로 변환되어 e에는 2만 저장된다. C언어에서 math.h에 들어있던 함수들은 대개 Math 클래스에서 찾아볼 수 있다.

while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

if (d > e) if 조건문이다. d가 e보다 크면 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

d = b; break; 반복문(while)을 탈출한다. 제어가 ❹번으로 간다.

mok = b / d; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.

nmg = b - mok * d;

if (nmg == 0) if 조건문이다. nmg가 0이면 break를 만나기 전까지 ❸~❺번 문장을 무한 반복한다.

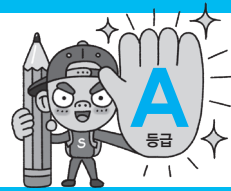
break; 반복문(while)을 탈출한다. 제어가 ❹번으로 간다.

```

    else
        d++; 수행 후 제어가 안쪽 while문의 시작점, 즉 ⑥번 윗 문장으로 이동한
            후 ⑥번부터 다시 시작한다.
    }
    c++;
    a[c] = d;
    if (b == d)
        break; 반복문(while)을 탈출한다. 제어가 ⑤번으로 간다.
    b = mok; 수행 후 제어가 바깥쪽 while문의 시작점, 즉 ⑤번 윗 문장으로 이동한 후 ⑤
        번부터 다시 시작한다.
}
for (int i = 0; i <= c; i++) 배열의 위치가 0부터 시작하기 때문에 첨자 i는
    0부터 소인수가 저장된 배열의 위치인 c가 될 때
    까지 반복하여 소인수를 출력한다.
    System.out.printf("%d ", a[i]);
    scanf.close();
}
}

```

022

진법변환 - 10진수를
2진수로 변환하기

선기 19.6, 07.4

문제

10진수를 입력 받아 2진수로 변환하는 순서도를 작성하시오. 단, 1000 이하의 숫자를 입력 받는다.

JAVA 코드

```
import java.util.Scanner;

public class Section022 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int b, bb, c, mok, nmg, i;
        ① int a[] = new int[10];
        ② b = scanf.nextInt();
        ③ bb = b;
        ④ c = -1; 배열의 위치가 0부터 시작하므로 배열 a의 첨자 c는 'c++(⑤)'을 수행한 후 0이 되도록
            -1로 초기화한다.
        do do~while 반복문의 시작점으로 ⑤~⑧, ⑪번 문장을 반복한다.
        {
            ⑤ c++;
            ⑥ mok = b / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는
                함수를 사용하지 않아도 된다.
            ⑦ nmg = b - mok * 2;
            ⑧ a[c] = nmg;
            ⑪ b = mok;
            ⑨ } while (mok != 0);
            ⑩ System.out.printf("%d ", bb);
            ⑫ for (i = c; i >= 0; i--) 배열의 위치가 0부터 시작하기 때문에 첨자 i는 2진수가
                저장된 마지막 위치인 c부터 0이 될 때까지 ⑬번을 반복
                수행한다.
            ⑬ System.out.printf("%d", a[i]);
                scanf.close();
        }
    }
}
```



JAVA 코드

```

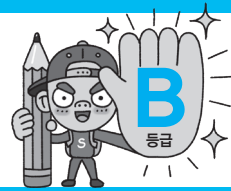
import java.util.Scanner;

public class Section022_Wait {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        int b, bb, c, mok, nmg, k, i;
        int a[] = new int[10];
        b = scanf.nextInt();
        bb = b;
        c = -1;  배열의 위치가 0부터 시작하므로 배열 a의 첨자 c는 'c++(6)'을 수행한 후 0이 되
                도록 -1로 초기화한다.
        do  do~while 반복문의 시작점으로 6~8, 10번 문장을 반복하여 수행한다.
        {
            5      c++;
            6      mok = b / 2;  정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를
                        구하는 함수를 사용하지 않아도 된다.
            7      nmg = b - mok * 2;
            8      a[9-c] = nmg;  배열의 위치가 0부터 시작하므로 배열 a의 뒤쪽은 a[9]부터 시
                        작한다. 배열의 뒤쪽부터 차례로 저장되도록 a[9-c]로 지정한
                        다.

            10     b = mok;
        } while (mok != 0);  mok가 0이 아니면 2진수 변환이 남았으므로 6번으로 가고,
                        아니면 2진수 변환을 모두 마친 것이므로 11번으로 간다.
        11     System.out.printf("%d ", bb);
        12     k = 9 - c;  배열의 위치가 0부터 시작하므로 2진수가 들어 있는 첫 번째 위치를 9-c로
                        계산한다.
        13     for (i = k; i < 10; i++)  배열의 위치가 0부터 시작하므로 반복 변수 i가 k
                        에서 9가 될 때까지 14번을 반복 수행한다.
        14         System.out.printf("%d", a[i]);
        scanf.close();
    }
}

```

023

진법변환 - 10진수를
임의의 진수로 변환하기

문제

10진수를 입력 받아 2진수, 8진수, 16진수로 변환해서 출력하는 순서도를 작성하시오. 단, 진수 표시에 사용될 0~9, A, B, C, D, E, F는 A[16] 배열에 저장되어 있다고 가정한다.

JAVA 코드

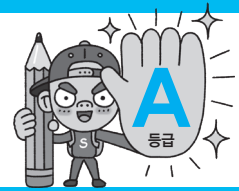
```
import java.util.Scanner;

public class Section023 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int b, c, d, e, f;
        ❶ char a[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A',
                    'B', 'C', 'D', 'E', 'F' }; 16개의 요소를 갖는 배열 A에 0~9, A~F까지 저장되어 있다고 가정하였으므로 값을 초기화한다.

        ❷ b = scanf.nextInt();
        ❷ c = scanf.nextInt();
        ❸ d = 1;
        ❹ while (d <= c) d가 c보다 작거나 같은 동안 ❺번 문장을 반복 수행한다.
        ❺     d *= b;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ❻ if (d > 1)
            ❼     d /= b;
            ❸ e = c / d;
            ❹ f = c - e * d;
            ❶ System.out.printf("%c", a[e]); 배열의 위치가 0부터 시작하므로 배열 a의 첨자로 구해진 몫을 그대로 사용하면 된다.

            ❶ if (d != 1)
            ❷     c = f; 수행 후 제어기가 while문의 시작점으로 이동하여 ❸번부터 다시 시작한다.
            else
                break; 반복문(while)을 탈출한다. 제어기가 ❹번으로 이동한다.
        }
        ❶ scanf.close();
    }
}
```

024 진법변환 - 소수점이 포함된 2진수를 10진수로 변환하기



산기 07.7, 기사 08.7

문제

10자리로 구성된 2진수를 입력 받아 10진수로 변환하여 출력하는 순서도를 작성하시오. 단, 10자리 2진수는 문자열로 되어 있고, 5번째 자리까지는 소수 이상이고, 6번째 자리부터 10번째 자리까지는 소수 이하를 의미한다.

JAVA 코드

```
import java.util.Scanner;

public class Section024 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ String a, munja;  JAVA에서는 문자열 변수를 사용할 수 있기 때문에, C언어에서와 같이 문자 배열을 사용하지 않는다. 변수 a에
                           입력된 각 자리를 분리해서 문자열로 만들기 위한 문자열 변수 munja도 선언한다.

        ❷ double d, e;
        ❸ a = scanf.nextLine();  키보드로부터 한 줄을 입력받아 문자열 형태로 a에 저장한다.
        ❹ double b = 0;
        ❺ int c = -1;  문자열의 위치가 0부터 시작하므로 문자열 변수 a의 첨자 c는 'c++(❹)'을 수행한 후 0이 되도록 -1로 초기화한다.
        while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ❻ c++;
            ❼ if (c <= 9)  문자열의 위치가 0부터 시작하므로 문자열 변수 a의 첨자 c는 0~9까지 10자리가 된다.
            {
                munja = Character.toString(a.charAt(c));  문자열 변수의 charAt() 메소드를 이용해 해당
                                                            위치의 문자를 가져오면 char 형태이므로 munja
                                                            에 저장할 수 없기 때문에 Character 클래스의
                                                            toString() 메소드를 이용해 문자열로 변환한다.

                ❽ d = Integer.parseInt(munja);  JAVA에서 C언어의 atoi() 함수를 대체할 수 있는 것으로 Integer 클
                                                            래스의 parseInt() 메소드를 이용한다. 문자열(String) 형태만 인수로
                                                            취할 수 있다는 점까지 atoi() 함수와 동일하다.

                ❾ e = d * Math.pow(2, 4 - c);  pow(2, 4-c)는 배열의 위치가 0부터 시작하기 때문이다. C언어에서 math.h
                                                            에 들어있던 함수들은 대개 Math 클래스에서 찾아볼 수 있다.

                ❿ b += e;  수행 후 제어가 while문의 시작점으로 이동하여 ❹번부터 다시 시작한다.
            }
            else
            {
                ❻ System.out.printf("%8.5f", b);
                break;  반복문(while)을 탈출한다. 제어가 ❿번으로 이동한다.
            }
        }
        ❿ scanf.close();
    }
}
```

025 최대값, 최소값 구하기



유형

1

최대값 구하기

문제

10개의 수치 자료를 입력 받아 배열에 저장한 후 저장된 자료 중 가장 큰 값을 찾는 순서도를 작성하시오.

JAVA 코드

```
import java.util.Scanner;

public class Section025_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);

        ❶ int i;
        ❶ int a[] = new int[10];
        ❷ int j = -1, max = 0;  배열의 위치가 0부터 시작하므로 배열 a의 첨자 j는 'j++(❸)'을
                           수행한 후 0이 되도록 -1로 초기화한다.

        do  do~while 반복문의 시작점으로 ❸, ❹번 문장을 반복 수행한다.
        {
            ❸ j++;
            ❺ a[j] = scanf.nextInt();
            ❹ } while (j < 9);  배열의 위치가 0부터 시작하므로 배열 a의 첨자 j는 0~9까지 10자리가
                           된다.

            ❻ for (i = 0; i <= 9; i++)  배열의 위치가 0부터 시작하므로 배열 a의 첨자 i는
                                   0~9까지 변하는 동안 ❶~❸번 문장을 반복 수행한다.
            {
                ❶ if (a[i] > max)
                ❷ max = a[i];
            }

            ❹ System.out.printf("%d", max);
            scanf.close();
        }
    }
}
```

유형

2

최대값, 최소값을 제외한 평점의 평균 구하기

상기 12.10

문제

7명의 채점 점수 중에서 최하위 점수와 최상위 점수를 제외한 5명 점수의 평균을 구하는 순서도를 작성하시오. 단, 7명의 채점 점수는 배열에 들어 있다.

JAVA 코드

```
import java.util.Scanner;

public class Section025_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int m, min, max, hap, avg, i;
        ① int a[] = new int[7];
        ② m = -1; 배열의 위치가 0부터 시작하므로 배열 a의 첨자 m은 'm++(⑨)'을 수행한 후 0이 되도록
            -1로 초기화한다.

        do
        {
            ③ m++;
            ④ a[m] = scanf.nextInt();
            ⑤ } while (m < 6); 배열의 위치가 0부터 시작하므로 배열 a의 첨자 m은 0~6까지 7자리가
                된다. m이 6보다 작은 동안 ③~④번 문장을 반복한다.
            ⑥ min = a[0]; 배열의 위치가 0부터 시작하므로 배열 a의 첫 번째 값은 a[0]이다.
            ⑦ max = a[0];
            ⑧ hap = a[0];
            ⑨ i = 0; 배열의 위치가 0부터 시작하므로 배열 a의 첨자 i는 'i++(⑭)'을 수행한 후 1이 되도록 0
                으로 초기화한다.
            ⑩ while(i < 6) 배열의 위치가 0부터 시작하므로 배열 a의 첨자 i는 0~6까지 7자리가 된다. i
                가 6보다 작은 동안 ⑪~⑯번 문장을 반복한다.

            {
                ⑭ i++;
                ⑮ hap += a[i];
                ⑯ if (a[i] < min)
                    min = a[i];
                ⑰ if (a[i] > max)
                    max = a[i];
            }
            ⑪ hap = hap - min - max;
            ⑫ avg = hap / 5;
            ⑬ System.out.printf("%d %d %d %d", min, max, hap, avg);
            scanf.close();
        }
    }
}
```

026 5의 배수의 개수와 합



문제 1부터 100까지의 수 중 5의 배수의 개수와 합을 출력하는 순서도를 작성하시오.

JAVA 코드

```

public class Section026 {
    public static void main(String[] args) {
        ① int cnt, hap, mok, nmg, i;
        ② cnt = hap = 0;
        ③ for (i = 1; i <= 100; i++)
        {
            ④ mok = i / 5;  정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는
                           함수를 사용하지 않아도 된다.
            ⑤ nmg = i - mok * 5;
            ⑥ if (nmg == 0) 나머지 0인지, 즉 5의 배수인지를 판별한다.
            {
                ⑦ cnt++;
                ⑧ hap += i;
            }
        }
        ⑨ System.out.printf("%d %d", cnt, hap);
    }
}

```

027 7에 가장 가까운 수 구하기



산기 20,11, 19,6, 18,10, 13,10, 10,10, 08,7, 07,11, 06,4

문제

9개의 1자리 양의 정수를 입력 받아 배열에 저장한 후 저장된 자료 중 7에 가장 가까운 자료를 찾는 순서도를 작성하시오. 단, 자료는 1자리 정수이고 근사값이 2개인 경우에는 나중에 찾은 값을 출력한다.

JAVA 코드

```
import java.util.Scanner;

public class Section027 {
    static final int FIND = 7; // 클래스 안에서 FIND는 항상 고정된 값 7을 갖도록 정의한다. 상수명 FIND는 사용자가 임의로 지정할 수 있다.

    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);

        ① int i, j, k, L, m = 0; // JAVA에서는 변수의 값이 정의되지 않고 사용되거나 사용될 가능성이 있는 경우 오류가 발생한다. ①번에서 L이 j보다 큰 경우 ③번을 수행하지 않고 ⑪번으로 진행되었을 때 정의되지 않은 알 수 없는 값이 m에 저장되어 있어 실행 시 오류가 발생하므로, c언어와 다르게 m을 0으로 초기화하였다.

        ① int a[] = new int[10];
        ② i = -1; // 배열의 위치가 0부터 시작하므로 배열 a의 첨자 i는 'i++(③)'을 수행한 후 0이 되도록 -1로 초기화한다.
        do
        {
            ③ i++;
            ⑤ a[i] = scanf.nextInt();
            ④ } while (i < 9); // 배열의 위치가 0부터 시작하므로 배열 a의 첨자 i는 0~9까지 10자리가 된다.
            ⑥ j = 9;
            ⑦ for (k = 0; k <= 9; k++) // 배열의 위치가 0부터 시작하므로 배열 a의 첨자 k는 0~9까지 변하는 동안 ③~⑤번 문장을 반복 수행한다.

            {
                ⑧ if (a[k] >= FIND) // FIND가 7로 정의되었으므로 배열의 해당 번째 요소가 7보다 크거나 같으면 ⑩번 문장을 실행하고, 아니면 ⑨번 문장으로 이동하여 실행을 계속한다.
                    ⑩ L = a[k] - FIND; // a[k]가 FIND, 즉 7보다 크거나 같은 경우로 a[k]에서 7을 뺀 값을 L에 저장한다. 제어가 ⑪번으로 이동한다.

                else
                    ⑨ L = FIND - a[k]; // a[k]가 FIND, 즉 7보다 작은 경우로 7에서 a[k]를 뺀 값을 L에 저장한다. 제어가 ⑪번으로 이동한다.
                ⑪ if (L <= j)
                {
                    ⑫ j = L;
                    ⑬ m = a[k]; // 수행 후 제어가 for문의 시작점으로 이동하여 ⑦번부터 다시 시작한다.
                }
            }

            ⑭ System.out.printf("%d", m);
            scanf.close();
        }
    }
}
```



JAVA 코드

```

public class Section027_Wait {
    public static void main(String[] args) {
        ❶ int m, n, x, s = 0;  JAVA에서는 변수의 값이 정의되지 않고 사용되거나 사용될
                               가능성이 있는 경우 오류가 발생한다. ❸번에서 m이 x보다
                               작거나 같은 경우 ❶번을 수행하지 않고 ❷번으로 진행되었
                               을 때 정의되지 않은 알 수 없는 값이 s에 저장되어 있어 실
                               행 시 오류가 발생하므로, C언어와 다르게 s를 0으로 초기
                               화하였다.

        ❶ int k[] = { 1,2,3,4,7,6,8,9,3,6 };  10개의 요소를 갖는 배열 k에 값
                                               이 저장되어 있다고 가정하였으므로 값을 초기화한다.

        ❷ m = 9;
        ❸ n = -1;  배열의 위치가 0부터 시작하므로 배열 k의 첨자 n은 'n++(❹)'을 수행한 후 0이 되
                   도록 -1로 초기화한다.

        do
        {
            ❹ n++;
            ❺ if (k[n] < 5)
            ❻ x = 5 - k[n];  수행 후 제어기가 ❸번으로 이동한다.
            else
            ❼ x = k[n] - 5;  수행 후 제어기가 ❸번으로 이동한다.
            ❽ if (m > x)
            {
                ❾ m = x;
                ❿ s = k[n];  수행 후 제어기가 do~while문의 시작점으로 이동하여 ❹번부
                           터 다시 시작한다.
            }
            11 } while (n < 9);  배열의 위치가 0부터 시작하므로 배열 k의 첨자 n은 0~9까지 10
                               자리가 된다.
            12 System.out.printf("%d", s);
        }
    }
}

```

028 보수 구하기



유형

1

1의 보수, 2의 보수 구하기

기사 09.10

문제

0 또는 1로 입력되는 다섯 개의 숫자를 배열에 입력받아 1의 보수와 2의 보수를 구하는 순서도를 완성하시오.

JAVA 코드

```
import java.util.Scanner;

public class Section028_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int a[] = new int[5];
        ① int b1[] = new int[5];
        ① int b2[] = new int[5];
        ① int i = -1, c = 1; 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 'i++(②)'을 수행한 후 0이 되도록 -1로 초기화한다.
        do
        {
            ② i++;
            ③ a[i] = scanf.nextInt();
            ④ b1[i] = 1 - a[i];
            ⑤ } while (i < 4); 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~4까지 변하는 동안 ②~④번 문장을 반복 수행한다.
        do
        {
            ⑥ b2[i] = b1[i] + c;
            ⑦ b2[i] = b2[i] % 2;
            ⑧ c = b1[i] * c;
            ⑨ i--;
            ⑩ } while (i >= 0); 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 4~0까지 변하는 동안 ⑥~⑨번 문장을 반복 수행한다.
            ⑪ for (int k = 0; k < 5; k++) 배열 a를 출력한다.
                System.out.printf("%d", a[k]);
                System.out.printf(" "); 배열 a를 출력한 후 한 칸" "을 띄운다.
                for (int k = 0; k < 5; k++) 배열 b1을 출력한다.
                    System.out.printf("%d", b1[k]);
                System.out.printf(" "); 배열 b1을 출력한 후 한 칸" "을 띄운다.
                for (int k = 0; k < 5; k++) 배열 b2를 출력한다.
                    System.out.printf("%d", b2[k]);
                scanf.close();
        }
    }
}
```

JAVA 코드

```

public class Section028_Type2 {
    public static void main(String[] args) {
        ❶ int i, j;
        ❶ int a[] = { 0,1,0,1,0 }; 5개의 요소를 갖는 배열 a에 값이 저장되어 있다고 가정
                                   하였으므로 값을 초기화한다.
        ❶ int b[] = new int[5];
        ❷ for (i = 4; i >= 0; i--) 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 4~0
                                   까지 변하는 동안 ❸~❹번 문장을 반복 수행한다.
        {
            ❸ if (a[i] == 1)
                break; 반복문(for)을 탈출한다. 제어가 ❺번으로 이동한다.
            ❹ b[i] = a[i];
        }
        ❺ b[i] = a[i];
        ❻ for (j = i - 1; j >= 0; j--) 배열의 위치가 0부터 시작하므로 배열의 첨자 j
                                   는 i-1부터 0이 될 때까지 ❼번 문장을 반복 수행
                                   한다.
        ❼ b[j] = 1 - a[j];
        ❸ for (int k = 0; k < 5; k++) 배열 a를 출력한다.
            System.out.printf("%d", a[k]);
            System.out.printf(" "); 배열 a를 출력한 후 한 칸(" ")을 띄운다.
            for (int k = 0; k < 5; k++) 배열 b를 출력한다.
            System.out.printf("%d", b[k]);
        }
    }
}

```

유형

3

보수의 보수 구하기

문제

2의 보수로 표현된 값의 2의 보수를 구하는 순서도를 완성하시오.

JAVA 코드

```
import java.util.Scanner;

public class Section028_Type3 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int a[] = new int[5];
        ① int b[] = new int[5];
        ① int i, c;
        ② for (i = 0; i <= 4; i++)    배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~4
                                   까지 변한다.
        ③     a[i] = scanf.nextInt();
        ④     c = 1;
        ⑤     for (i = 4; i >= 0; i--)    배열의 위치가 0부터 시작하므로 배열의 첨자 i는 4~0
                                   까지 변한다.
        {
            ⑥     b[i] = a[i] - c;
            ⑦     if (a[i] == 0 && c == 1)    현재 위치의 값(a[i])이 0 그리고(&&) 현재의 감수(c)
                                           가 1인 경우에는 ⑨번 문장을 실행하고 아니면 ⑧
                                           번 문장으로 이동하여 실행을 계속한다.
            ⑨         c = 1;
            else
            ⑧         c = 0;
            ⑩         b[i] = Math.abs(b[i]);    C언어에서 math.h에 들어있던 함수들은 대개
                                           Math 클래스에서 찾아볼 수 있다.
        }
        ⑪     for (i = 0; i <= 4; i++)    배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~4
                                   까지 변한다.
        ⑫         b[i] = 1 - b[i];
        ⑬     for (int k = 0; k <= 4; k++)    배열 a를 출력한다.
            System.out.printf("%d", a[k]);
            System.out.printf(" ");    배열 a를 출력한 후 한 칸(" ")을 띄운다.
            for (int k = 0; k <= 4; k++)    배열 b를 출력한다.
                System.out.printf("%d", b[k]);
            scanf.close();
        }
    }
}
```

029 그레이 코드 변환하기



기사 18.10, 14.4, 07.11

문제

0 또는 1로 입력되는 다섯 개의 숫자를 배열에 입력 받아 그레이 코드 또는 이진수로 변환하는 순서도를 완성하십시오.

JAVA 코드

```
import java.util.Scanner;

public class Section029 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int i;
        ① int input[] = new int[5];
        ① int cont[] = new int[4];
        ② for (i = 0; i <= 4; i++) 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~4
                               까지 변환다.
        ③     input[i] = scanf.nextInt();
        ④     cont[0] = input[1]; 배열의 위치가 0부터 시작하므로 입력받은 배열의 두 번째 요소
                               는 input[1]이고 변환되는 배열의 첫 번째 요소는 cont[0]이
                               된다.
        ⑤     if (input[0] == 1) 배열의 위치가 0부터 시작하므로 입력받은 배열의 첫 번째 요소
                               가 input[0]이 된다.
        {
            ⑥     for (i = 0; i <= 2; i++) 배열의 위치가 0부터 시작하므로 배열의 첨자 i
                               는 0~2까지 변환다.
            {
                ⑦         if (input[i + 2] == cont[i])
                ⑧             cont[i + 1] = 0; 수행 후 제어가 for문의 시작점으로 이동하여
                               ⑥번부터 다시 시작한다.
                else
                ⑨             cont[i + 1] = 1; 수행 후 제어가 for문의 시작점으로 이동하여
                               ⑥번부터 다시 시작한다.
            }
        }
        else
        {
            ⑩     for (i = 0; i <= 2; i++) 배열의 위치가 0부터 시작하므로 배열의 첨자 i
                               는 0~2까지 변환다.
            {
                ⑪         if (input[i + 1] == input[i + 2])
                ⑫             cont[i + 1] = 0; 수행 후 제어가 for문의 시작점으로 이동하여
                               ⑩번부터 다시 시작한다.
            }
        }
    }
}
```

```

        else
            cont[i + 1] = 1;
    }
}
if (input[0] == 1)
    System.out.printf("이진수 ");
else
    System.out.printf("그레이코드 ");
for (i = 0; i <= 3; i++)
    System.out.printf("%d", cont[i]);
scanf.close();
}

```

⑬ 수행 후 제어기가 **for**문의 시작점으로 이동하여 ⑩번부터 다시 시작한다.

⑭ 배열의 위치가 0부터 시작하므로 입력받은 배열의 첫 번째 요소가 `input[0]`이 된다.

⑮ 수행 후 제어기가 ⑦번으로 이동한다.

⑯ 수행 후 제어기가 ⑦번으로 이동한다.

⑰ 배열의 위치가 0부터 시작하므로 배열의 첨자 `i`는 0~3까지 변한다.

⑱

030 큰 수 더하기



유형

1

십진수 더하기

문제

100건 이내의 12자리로 구성된 숫자를 더하는 순서도를 작성하시오. 단, 12자리의 숫자는 각 자리가 분리되어 배열에 입력된다. 예를 들어 999,999,999,999라면

9	9	9	9	9	9	9	9	9	9	9	9
---	---	---	---	---	---	---	---	---	---	---	---

와 같이 입력된다. 단, 배열의 첫 번째 요소로 0을 입력 받으면 계산 후 결과를 출력하고 프로그램을 종료한다. 단, 결과값이 들어갈 배열에는 초기 값으로 0이 들어 있다고 가정한다.

JAVA 코드

```

import java.util.Scanner;

public class Section030_Type1 {
    public static void main(String[] arags) {
        Scanner scanf = new Scanner(System.in);
        ① int i, j, mok, nmg;
        ① int a[] = new int[12];
        ① int b[] = new int[14];  Java에서 객체를 선언할 때 특별한 값을 저장하지 않으면
                               정수형인 경우 0으로 초기화된다. 배열 또한 객체이므로 따
                               로 0으로 초기화하는 과정이 필요 없다.
        while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
                       지 { } 사이의 문장을 무한 반복한다.
        {
            ② for (i = 0; i <= 11; i++)  배열의 위치가 0부터 시작하므로 배열의 첨자 i
                                       는 0~11까지 변한다.
            ③ a[i] = scanf.nextInt();
            ④ if (a[0] == 0)  배열의 위치가 0부터 시작하므로 배열 a의 첫 번째 요소는 a[0]이
                           된다.
                break;  반복문(while)을 탈출한다. 제어가 ⑦번으로 이동한다.
            else
            ⑤ for (i = 0; i <= 11; i++)  배열의 위치가 0부터 시작하므로 배열의
                                       첨자 i는 0~11까지 변한다.
            ⑥ b[i + 2] += a[i];
            }
            ⑦ for (i = 13; i >= 1; i--)  배열의 위치가 0부터 시작하므로 배열의 첨자 i는
                                       13~1까지 변한다.
            {
            ⑧ mok = b[i] / 10;  정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를
                               구하는 함수를 사용하지 않아도 된다.
            ⑨ nmg = b[i] - mok * 10;
            ⑩ b[i] = nmg;
            ⑪ b[i - 1] += mok;
            }
            ⑫ if (b[0] == 0)  배열의 위치가 0부터 시작하므로 배열 b의 첫 번째 요소는 b[0]이 된다.
            ⑬ j = 1;  배열의 위치가 0부터 시작하므로 배열 b의 두 번째 요소를 가리키는 첨자는 1이
                       된다.
            else
            ⑭ j = 0;  배열의 위치가 0부터 시작하므로 배열 b의 첫 번째 요소를 가리키는 첨자는 0이 된다.
            ⑮ for (i = j; i <= 13; i++)  배열의 위치가 0부터 시작하므로 배열의 첨자 i는 j부
                                       터 시작하여 13이 될 때까지 반복 수행한다.
            ⑯ System.out.printf("%d", b[i]);
                scanf.close();
            }
        }
    }
}

```

기사 09.7

문제

배열 $X(10)$ 와 $Y(10)$ 에 이진수가 각각 입력되어 있다. 두 이진수의 덧셈 결과를 이진수 형태로 출력하는 순서도를 작성하되 덧셈의 결과 MSB(최상위 비트)에서 올림수가 발생하면 "OVERFLOW!!"라고 출력한다.

JAVA 코드

```

import java.util.Scanner;

public class Section030_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int i, z, c, j;
        ❶ int x[] = new int[10];
        ❶ int y[] = new int[10];
        ❶ int a[] = new int[10];
        for (i = 0; i <= 9; i++) 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~9
            까지 변한다.
            x[i] = scanf.nextInt();
        for (i = 0; i <= 9; i++) 배열의 위치가 0부터 시작하므로 배열의 첨자 i는 0~9
            까지 변한다.
            y[i] = scanf.nextInt();
        ❷ z = c = 0;
        ❷ j = 9; 배열의 위치가 0부터 시작하므로 배열의 끝 위치는 9이다. 배열의 끝 위치를 지정할 첨자
            j를 9로 초기화한다.
        do
        {
            ❸ z = x[j] + y[j] + c;
            ❹ if (z < 2)
            {
                ❷ c = 0;
                ❸ a[j] = z; 수행 후 제어가 ❸번으로 이동한다.
            }
            else
            {
                ❺ c = 1;
                ❻ a[j] = z - 2; 수행 후 제어가 ❹번으로 이동한다.
            }
            ❾ j--;
            ❿ } while (j >= 0); 배열의 위치가 0부터 시작하므로 배열의 첨자 j는 9~0까지 변한다.
            11 if (c == 0)
            {
                do
                {
                    13 j++;
                    14 System.out.printf("%d", a[j]);
                    15 } while (j < 9); 배열의 위치가 0부터 시작하므로 배열의 첨자 j는 0~9까지 변한
                        다. 수행 후 제어가 15번으로 이동한다.
                }
                else
                {
                    12 System.out.printf("OVERFLOW!!"); 수행 후 제어가 16번으로 이동한다.
                    16 scanf.close();
                }
            }
        }
    }
}

```



문제 7 다음은 99를 넘지 않는 소수의 개수를 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test07 {  
    static int isprime(int number) {  
        ④ int i; 정수형 지역 변수 i를 선언한다.  
        ⑤ for(i = 2; i <= number; i++) 반복 변수 i가 2에서 시작하여 1씩 증가하면서  
            number보다 작거나 같을때까지 ⑥번을 반복한다.  
        ⑥ if(number % i == 0) number를 i로 나눈 나머지가 0인지를 확인한다.  
        ⑦ return 0; 나머지가 0이면 반환값 0을 가지고 isprime() 함수를 호출했던 곳으로 제  
            어를 옮긴다.  
        ⑧ return 1; 반환값 1을 가지고 isprime() 함수를 호출했던 곳으로 제어를 옮긴다.  
    }  
  
    public static void main(String[] args) {  
        ① int number = 99, cnt = 0, i; 정수형 변수 number, cnt, i를 선언하고 number  
            에는 100을 cnt에는 0을 할당한다.  
        ② for(i = 2; i <= number; i++) 반복 변수 i가 2에서 시작하여 1씩 증가하면서  
            number(99)보다 작거나 같을때까지 ③번을 반복  
            한다. 즉 ③번 문장을 98회 반복한다.  
        ③ cnt = cnt + isprime(i); i의 값을 인수로 하여 isprime() 함수를 호출한 다음 돌려받  
            은 값을 cnt와 더한 후 그 값을 다시 cnt에 저장한다.  
        ④ System.out.printf("%d를 넘지 않는 소수는 %d개입니다.\n", number, cnt);  
            number와 cnt의 값을 출력 서식에 맞게 출력한 후 커서를 다음 줄 처음으로 옮긴다.  
    }  
}
```



문제 8

기사 17년 4월

다음은 최소값을 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.util.Scanner;
public class Test08 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int num[] = new int[10];           입력 받은 숫자가 저장될 배열 num을 선언한다.
        ❷ int min = 9999;                   최소값이 저장될 변수 min을 선언하고 초기값으로 9999를 할당한다.
        ❸ int i;                           변수 i를 선언한다.
        ❹ for (i = 0; i < 10; i++) {        반복 변수 i가 0에서 시작하여 1씩 증가하면서 10보다 작은 동안 ❹번을 반복하여 수행한다.
            ❺ num[i]=scanf.nextInt();      최소값을 판단할 숫자를 키보드로 입력받는다.
        }
        ❻ for (i = 0; i < 10; i++) {        반복 변수 i가 0에서 시작하여 1씩 증가하면서 10보다 작은 동안 ❷번을 반복하여 수행한다.
            ❼ if (min > num[i]) {           min이 num[i]보다 크면 ❸번으로 가고, 아니면 ❹번으로 간다.
                ❽ min = num[i];           num[i]의 값을 min에 저장한다.
            }
        }
        ❹ System.out.printf("가장 작은 값은 %d이다.", min); min의 값을 출력하고 끝낸다.
    }
}
```



상기 18년 6월

문제 9 다음은 1부터 100의 범위 안에서 가장 큰 소수를 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

1  import java.lang.Math;      sqrt() 함수가 정의되어 있는 헤더 파일이다.

public class Test {
    public static void main(String[] args) {
2      int p = 2;      정수형 변수 p를 선언하고 2로 초기화한다.
3      int n = 3;      정수형 변수 n을 선언하고 3으로 초기화한다.
4      while (true) {      조건을 만족하는 동안 반복하는데, 조건이 true, 즉 참이므로 무한
                            반복한다. 결국 9번이나 번의 조건을 만족하여 break를 만나기 전
                            까지 5~6 사이의 문장을 반복하여 수행한다.
5          double t = Math.sqrt(n);      n의 제곱근을 구한 후 실수형 변수 t에 저장한다.
6          int m = (int)t;      실수형 변수 t의 값을 정수형으로 변환한 후 정수형 변수 m
                                에 저장한다.
7          for (int i = 2; i <= m; i++) {      반복 변수 i가 2에서 시작하여
                                                1씩 증가하면서 m보다 작거나
                                                같은 동안 8~12 사이의 문장을
                                                반복한다.
8              int r = n % i;      n을 i로 나눈 나머지를 정수형 변수 r에 저장한다.
9              if (r == 0)      r이 0이면 10번으로 가고 아니면 11번 문장을 실행
                                한다.
10                 break;      반복문(while)을 탈출한다. 제어가 15번으로 이동
                                한다.
11                 if (i == m)      i와 m이 같으면 12번으로 가고 아니면 7번부터 다시
                                시작한다.
12                     p = n;      p에 소수 n을 저장한다.
                            }
13         n++;      'n = n + 1'과 동일하다. n의 값을 1씩 누적시킨다.
14         if (n > 100)      n이 100보다 크면 15번으로 가고 아니면 4번부터 다시 시
                                작한다.
15             break;      반복문(while)을 탈출한다. 제어가 16번으로 이동한다.
        }
16     System.out.printf("%d\n", p);      결과 97
    }
}

```



문제 10 다음은 정수 N을 입력받아 소인수 분해하여 그 결과를 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.util.Scanner;

public class Test10 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int n, c, k, y; 정수형 변수 n, c, k, y와 소인수를 저장할 배열 s를 선언한다.
        int s[] = new int[20];
        while (true)
        {
            ② n = scanf.nextInt(); 소인수로 분해할 숫자를 입력 받는다.
            ③ if (n < 2) n이 2보다 작으면 소인수 분해 과정을 모두 마친 것으로 프로그램을 종료하고 아니면 ④번으로 간다.
                break;
            ④ c = -1; 배열의 위치가 0부터 시작하므로 인수를 저장할 배열의 위치 c는 'c++(⑧)'을 수행한 후 0이 되도록 -1로 초기화
                한다.

            do
            {
                ⑤ k = 2; 입력 받은 수 n을 2부터 나눠보기 위해 k를 2로 초기화한다.
                ⑥ while (n % k != 0) n을 k로 나눈 나머지가 0이 아니면 다음 수로 나눠보기 위해 ⑦번으로 가고, 아니면
                    소인수이므로 ⑧번으로 간다.
                ⑦ k++; n을 다음 수로 나눠보기 위해 k를 1 증가시킨다.
                ⑧ c++; 소인수의 개수를 세기 위해 c를 1 증가시킨다.
                ⑨ s[c] = k; s[c]에 소인수 k를 저장한다.
                ⑩ n /= k; 다음 소인수를 구하기 위해 n을 k로 나눈 몫을 다시 n에 저장한다.
                ⑪ } while (n != 1); n이 1이 아니면 소인수 분해가 남은 것이므로 ⑤번으로 가고, 아니면 소인수 분해 과정을 모두
                    마친 것이므로 ⑫번으로 간다.
                ⑫ if (c == 0) 배열의 위치가 0부터 시작하므로 c가 0이면 "소수"이므로 ⑬번으로 가고, 아니면 소인수를 출력하기 위
                    해 ⑭번으로 간다.
                ⑬ System.out.printf("소수"); "소수"를 출력하고 소인수로 분해할 새로운 숫자를 입력받기 위해 ②번
                    으로 간다.

                else
                {
                    ⑭ for (y = 0; y <= c - 1; y++) 배열의 위치를 나타내는 반복 변수 y가 0에서 c-1이 될 때까지 1
                        씩 증가하면서 ⑮번을 반복하여 수행한다.
                    ⑮ System.out.printf("%dX", s[y]); s 배열에 저장된 값을 "X" 기호와 함께 출력한다.
                    ⑯ System.out.printf("%d", s[c]); s 배열에 저장된 마지막 값은 "X" 기호 없이 값만 출력한다.
                }
            }
            scanf.close();
        }
    }
}
```



산기 18년 6월

문제 11 다음은 부호를 포함하여 8비트로 표현된 2진수 값을 10진수로 변환하여 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

public class Test11 {
    public static void main(String[] args) {
        ① int dec, p, k; 정수형 변수 dec, p, k와 부호를 포함한 2진수가 저장되어 있는 배열
           s를 선언한다.

        int s[] = { 0, 0, 0, 1, 1, 1, 1, 1 };
        ② dec = 0; dec는 10진수의 합계가 누적되므로 0으로 초기화한다.
        ③ p = 1; 배열의 위치가 0부터 시작하므로 p는 부호를 제외한 두 번째 위치를 지정해야 하기
           때문에 1로 초기화한다.

        do
        {
            ④ k = (int) Math.pow(2, (7 - p)); 2진수 각 자리에 해당하는 고유
               자리값을 10진수로 변환한다. 첫 번째 자리는 2의 6승으로 64인데,
               배열의 위치가 0부터 시작하므로 6승을 만들기 위해 지수를 (7-p)
               로 지정한다.
               C언어에서 math.h에 들어있던 함수들은 대개 Math 클래스에서
               찾아볼 수 있다. JAVA에서 pow의 반환값은 double이므로 int
               로 바꿔줘야 한다.

            ⑤ dec += k * s[p]; 2진수 각 자릿수에 해당하는 값을 10진수로 변환하여 dec
               에 누적한다.

            ⑥ p++; p를 1씩 증가시킨다.
            ⑦ } while (p < 8); 배열의 위치가 0부터 시작하므로 p가 8보다 작으면 아직 10진수
               변환이 끝나지 않은 것이므로 ④번으로 가고, 아니면 ⑧번으로 간
               다.
            ⑧ if (s[0] != 0) 배열의 위치가 0부터 시작하므로 s 배열의 첫 번째 요소는 s[0]이 된
               다. s[0]이 0이 아니면 음수로 표현된 2의 보수이므로 ⑨번으로 가고, 아
               니면 배열 s의 값을 출력하기 위해 ⑩번으로 간다.

            {
                ⑨ dec = 128 - dec; 2의 보수로 표현된 수치이므로 다시 2의 보수를 취해 10진
                   수 값을 구한다. 7자리의 2진수 이므로 2의 보수를 구하려
                   면 100000000(128)에서 해당 수치를 빼면 된다.

                ⑩ dec *= -1; dec에 (-1)을 곱해 음수로 표현한다.
            }
            ⑪ System.out.printf("%d", dec); dec를 출력하고 끝낸다.
        }
    }
}

```



문제 12 다음은 두 수를 입력받아 두 수 사이의 값들의 합계와 평균을 구하여 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.util.Scanner;

public class Test12 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int a, b, n; 정수형 변수 a, b, n을 선언한다.
        int sum = 0, cnt = 0, avg = 0; cnt는 1씩 누적되며 증가하는 변수이므로 0으로 초기화하고 합계가 저장될 sum과 평균이 저장될 avg는 0으로 초기화한다.

        ❷ a = scanf.nextInt(); 두 수를 입력받는다.
        b = scanf.nextInt();

        ❸ if (a > b) a가 b보다 크면 ❹번으로 가고, 아니면 ❷번으로 간다.
        ❹ for (n = b; n <= a; n++) n이 b부터 a가 될 때까지 1씩 증가하면서 ❺~❻번을 반복하여 수행한다.
        {
            ❺ sum += n; n을 합계 sum에 누적한다.
            ❻ cnt++; 수의 개수를 구하기 위해 cnt를 1씩 증가시킨다.
        }
        else
        ❷ for (n = a; n <= b; n++) n이 a부터 b가 될 때까지 1씩 증가하면서 ❸~❹번을 반복하여 수행한다.
        {
            ❸ sum += n; n을 합계 sum에 누적한다.
            ❹ cnt++; 수의 개수를 구하기 위해 cnt를 1씩 증가시킨다.
        }
        ❺ avg = sum / cnt; 평균을 구한다.
        ❻ System.out.printf("%d %d", sum, avg); sum과 avg를 출력하고 끝낸다.

        scanf.close();
    }
}
```



문제 13 다음은 100에서 500을 포함하여 두 수 사이의 자연수 중 홀수의 합, 짝수의 합, 10의 배수의 합을 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test13 {  
    public static void main(String[] args) {  
        ❶ int num; 정수형 변수 num을 선언하고 홀수의 합, 짝수의 합, 10의 배수의 합이 누적될 변수를 각각 0으로 초기화한다.  
        int totodd = 0, toteven = 0, tot10su = 0;  
        ❷ num = 99; num은 100부터 시작하는데, ❸번에서 1 증가시키므로 99로 초기화한다.  
        while (true) { while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.  
            ❸ num++; num을 100부터 500까지 증가시키기 위해 1씩 증가시킨다.  
            ❹ if (num > 500) { num이 500보다 크면 계산 작업을 마쳤으므로 ❶번으로 가고, 아니면 다음을 수행하기 위해 ❸번으로 간다.  
                10 System.out.printf("%d %d %d", totodd, toteven, tot10su); totodd, toteven, tot10su를 출력한다.  
                break; 반복문(while)을 탈출하여 프로그램을 종료한다.  
            }  
            else  
            {  
                ❺ if (num % 2 == 0) { 2로 나눈 나머지가 0이면 짝수, 아니면 홀수이다. num을 2로 나눈 나머지가 0이면 ❷번으로 가고, 아니면 ❻번으로 간다.  
                    ❷ toteven += num; 짝수의 합계를 누적한다.  
                    ❸ if (num % 10 == 0) 10으로 나눈 나머지가 0이면 10의 배수이다. 10의 배수이면 ❹번으로 가고, 아니면 ❸번으로 간다.  
                        9 tot10su += num; 10의 배수의 합계를 누적한다.  
                    }  
                    else  
                    ❻ totodd += num; 홀수의 합계를 누적한다.  
                }  
            }  
        }  
    }  
}
```



문제 14 다음은 1에서 100까지의 정수 중에서 3의 배수이면서 5의 배수인 수를 구하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test14 {  
    public static void main(String[] args) {  
        ❶      int i = 0, su1 = 0, su2 = 0, su3 = 0;   i는 1씩 누적되며 증가하는  
                                                       변수이므로 0으로 초기화한다. 3의 배수, 5의 배수, 3과 5의 배수  
                                                       인지 판별하는 나머지 값이 각각 저장될 변수 su1, su2, su3를  
                                                       0으로 초기화한다.  
        ❷      while (i != 100)   i가 100이 아니면 계속 처리하기 위해 ❸~❺번을 반복하여 수행  
                                   하고, 아니면 모든 처리를 마친 것이므로 끝낸다.  
        {  
            ❸      i++;   i를 1부터 100까지 증가시키기 위해 1씩 증가시킨다.  
            ❹      su1 = i % 3;   i를 3으로 나눈 나머지를 su1에 저장한다.  
            ❺      su2 = i % 5;   i를 5로 나눈 나머지를 su2에 저장한다.  
            ❻      su3 = su1 + su2;   su1과 su2를 더하여 su3에 저장한다. su3은 i가 3의  
                                   배수이면서 5의 배수인지를 확인하기 위한 값이다.  
            ❼      if (su3 == 0)   su3이 0이면 현재 i값이 3의 배수이면서 5의 배수이므로 ❸번  
                                   으로 가고, 아니면 다음 수를 확인하기 위해 ❷번으로 간다.  
            ❽      System.out.printf("%d ", i);   i를 출력하고 다음을 수행하  
                                                       기 위해 ❷번으로 간다.  
        }  
    }  
}
```

031 선택 정렬



선기 18.4, 05.10, 기사 19.6, 18.4, 12.7, 08.4

문제

10개의 수치 자료를 입력 받아 배열에 저장한 후 저장된 자료를 오름차순으로 정렬하는 순서도를 작성하시오.

JAVA 코드

```
import java.util.Scanner;

public class Section031 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int m, i, j, k, x;
        ① int data[] = new int[10];
        ② m = -1; 배열의 위치가 0부터 시작하므로 배열 data의 첨자 m에 'm++(⑤)'을 수행한 후 0이 되도록 -1로 초기화한다.
        do ③ do~while 반복문의 시작점으로 ③~④번을 반복하여 수행한다.
        {
            ③ m++;
            ④ data[m] = scanf.nextInt();
            ⑤ } while (m < 9); 배열의 위치를 0부터 시작하기 때문에 첨자의 최종값이 순서도와 다르다.
            배열 첨자 m은 9보다 작거나 같은 동안 ③~④번을 반복 수행한다.
        ⑥ i = -1; 배열 data의 첨자 i가 'i++(⑦)'을 수행한 후 0이 되도록 -1로 초기화한다.
        do ⑦ do~while 반복문의 시작점으로 ⑦~⑩번을 반복하여 수행한다.
        {
            ⑦ i++;
            ⑧ j = i;
            do ⑨ do~while 반복문의 시작점으로 ⑨~⑩번을 반복하여 수행한다.
            {
                ⑨ j++;
                ⑩ if (data[i] > data[j])
                {
                    ⑪ k = data[i];
                    ⑫ data[i] = data[j];
                    ⑬ data[j] = k;
                }
                ⑭ } while (j < 9); j는 9보다 작은 동안 ⑨~⑩번 문장을 반복 수행한다.
            } while (i < 8); i는 8보다 작은 동안 ⑦~⑩번 문장을 반복 수행한다.
            ⑮ for (x = 0; x <= 9; x++) 배열의 위치가 0부터 시작하므로 x가 0에서 9까지 변하는 동안 배열의 값을 반복 출력한다.
            ⑮ System.out.printf("%d ", data[x]);
            scanf.close();
        }
    }
}
```



JAVA 코드

```

import java.util.Scanner;

public class Section031_Wait {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int m, i, j, k, x;
        int data[] = new int[10];
        ❷ for (m = 0; m <= 9; m++) 배열 data의 위치가 0부터 시작하므로 첨자 m은
                                0에서 9까지 변하는 동안 반복 수행한다.
        ❸     data[m] = scanf.nextInt();
        ❹     for (i = 0; i <= 8; i++) 배열 data의 첨자 i는 0에서 8까지 변하는 동안
                                ❶~❸번을 반복 수행한다.
        {
        ❺         for (j = i + 1; j <= 9; j++) 배열 data의 첨자 j는 i+1부터 9
                                까지 변하는 동안 ❸~❹번을 반복
                                수행한다.
            {
        ❻             if (data[i] > data[j])
            {
        ❼                 k = data[i];
        ❽                 data[i] = data[j];
        ❾                 data[j] = k;
            }
        }
        }
        ❿ for (x = 0; x <= 9; x++) 배열의 위치가 0부터 시작하므로 x가 0에
                                서 9까지 변하는 동안 배열의 값을 반복 출력
                                한다.
        ⓫     System.out.printf("%d ", data[x]);
        scanf.close();
    }
}

```

032 버블 정렬



유형

1

버블 정렬

상기 14.7, 08.4

문제

배열에 기억된 10건의 자료를 오름차순으로 정렬하는 순서도를 작성하시오.

JAVA 코드

```

import java.util.Scanner;

public class Section032_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);

        ① int n, i, j, k;
        ① int data[] = new int[10];
        ② n = -1; 배열의 위치가 0부터 시작하므로 배열 data의 첨자 n이 'n++(③)'을 수행한 후 0이 되도록 -1로 초기화한다.
        do do~while 반복문의 시작점으로 ③~④번을 반복하여 수행한다.
        {
            ③ n++;
            ④ data[n] = scanf.nextInt();
            ⑤ } while (n < 9);
            ⑥ i = 0; 배열 data의 첨자 i가 'i++(⑦)'를 수행한 후 10이 되도록 0으로 초기화한다.
            do do~while 반복문의 시작점으로 ⑦~⑯번을 반복하여 수행한다.
            {
                ⑦ i++;
                ⑧ j = -1;
                do do~while 반복문의 시작점으로 ⑨~⑯번을 반복하여 수행한다.
                {
                    ⑨ j++;
                    ⑩ if (data[j] > data[j + 1])
                    {
                        ⑪ k = data[j];
                        ⑫ data[j] = data[j+1];
                        ⑬ data[j+1] = k;
                    }
                    ⑭ } while (j < 9 - i); j는 9-i보다 작은 동안 ⑨~⑯번을 반복 수행한다.
                } while (i < 9); i는 9보다 작은 동안 ⑦~⑯번을 반복 수행한다.
                ⑮ for (int x = 0; x <= 9; x++) 배열의 위치가 0부터 시작하므로 x가 0에서 9까지 변화하는동안 배열의 값을 반복 출력한다.

                System.out.printf("%d ", data[x]);
                scanf.close();
            }
        }
    }
}

```



JAVA 코드

```

import java.util.Scanner;

public class Section032_Wait {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int n, i, j, k;
        ① int data[] = new int[10];
        ② n = -1; 배열의 위치가 0부터 시작하므로 배열 data의 첨자 n이 'n++(③)'를 수행한 후
           0이 되도록 -1로 초기화한다.
        do ③~while 반복문의 시작점으로 ③~④번을 반복하여 수행한다.
        {
            ③ n++;
            ④ data[n] = scanf.nextInt();
            ⑤ } while (n < 9);
            ⑥ for (i = 1; i <= 9; i++) i는 배열 첨자로 사용되지 않으므로 0부터 시작하
               지 않아도 된다.
            {
                ⑦ for (j = 0; j <= 9 - i; j++) 배열의 위치는 0부터 시작하고, j는
                   배열의 첨자로 사용되므로 j는 0부터 9-i까지 반
                   복 수행한다.
                {
                    ⑧ if (data[j] > data[j + 1])
                    {
                        ⑨ k = data[j];
                        ⑩ data[j] = data[j + 1];
                        ⑪ data[j + 1] = k;
                    }
                }
            }
            ⑫ for (int x = 0; x < 10; x++) 배열의 위치가 0부터 시작하므로 x가 0에
               서 9까지 변하는 동안 배열의 값을 반복 출
               력한다.
                System.out.printf("%d ", data[x]);
                scanf.close();
            }
        }
    }
}

```

기사 13.4, 11.7, 06.4

문제

배열에 기억된 10건의 자료를 오름차순으로 정렬하는 순서도를 작성하시오. 단, 정렬 수행중 특정 회전에서 정렬을 위한 교환이 한 번도 이루어지지 않으면 정렬이 완료된 것이므로 그 때까지의 교환 횟수와 정렬된 자료를 출력하고 끝낸다.

JAVA 코드

```
import java.util.Scanner;

public class Section032_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int n, i, j, sw, cnt, k;
        ① int data[] = new int[10];
        ② n = -1; 배열의 위치가 0부터 시작하므로 배열 data의 첨자 n이 'n++(③)'를 수행한 후 0이 되도록 -1로 초기화한다.
        do do~while 반복문의 시작점으로 ③~④번을 반복하여 수행한다.
        {
            ③ n++;
            ④ data[n] = scanf.nextInt();
            ⑤ } while (n < 9);
            ⑥ cnt = 0;
            ⑦ for (i = 1; i <= 9; i++)
            {
                ⑧ sw = 0;
                ⑨ for (j = 0; j <= (9 - i); j++) 배열의 위치가 0부터 시작하고, j는 배열의 첨자로 사용되므로 j는 0부터 9-i까지 반복 수행한다.
                {
                    ⑩ if (data[j] > data[j + 1])
                    {
                        ⑪ k = data[j];
                        ⑫ data[j] = data[j + 1];
                        ⑬ data[j + 1] = k;
                        ⑭ cnt++;
                        ⑮ sw = 1;
                    }
                }
            }
            ⑯ if (sw == 0)
                break; for 반복문을 탈출한다. 제어가 ⑰번으로 이동한다.
        }
        ⑰ System.out.printf("%d ", cnt);
        for (int x = 0; x <= 9; x++) 배열의 위치가 0부터 시작하므로 x가 0에서 9까지 변하는동안 배열의 값을 반복 출력한다.
            System.out.printf("%d ", data[x]);
        scanf.close();
    }
}
```

유형

3

버블 정렬 3 – 좌우로 번갈아 가면서 정렬

기사 08.11

문제

버블 정렬 기법을 응용하여 한 번은 왼쪽에서 오른쪽으로 진행하면서 최대값을 우측으로 보내고, 한 번은 오른쪽에서 왼쪽으로 진행하면서 최소값을 좌측으로 보내는 방법을 반복하면서 정렬하는 순서도를 작성하시오.

JAVA 코드

```

public class Section032_Type3 {
    public static void main(String[] args) {
        ① int n, i, shift = 0, left, right, buf;
        ② int d[] = {8,5,6,2,4,1,3,7,9,10};
        ③ n = 9;
        ④ left = 0;
        ⑤ right = n;
        ⑥ while (left < right)
        {
            ⑦ for (i = left; i <= right - 1; i++)
            {
                ⑧ if (d[i] > d[i + 1])
                {
                    ⑨ buf = d[i];
                    ⑩ d[i] = d[i + 1];
                    ⑪ d[i + 1] = buf;
                    shift = i;
                }
            }
            ⑫ right = shift;
            ⑬ for (i = right; i >= left + 1; i--)
            {
                ⑭ if (d[i - 1] > d[i])
                {
                    ⑮ buf = d[i - 1];
                    ⑯ d[i - 1] = d[i];
                    ⑰ d[i] = buf;
                    ⑱ shift = i;
                }
            }
            ⑲ left = shift;
        }
        ⑳ for (int x = 0; x < 10; x++)
            System.out.printf("%d ", d[x]);
    }
}

```

JAVA에서는 변수의 값이 정의되지 않고 사용되거나 사용될 가능성이 있는 경우 오류가 발생한다. ⑥번에서 `left`가 `right-1`보다 클 경우 ⑩번을 수행하지 않고 ⑫번으로 진행되었을 때 정의되지 않은 알 수 없는 값이 `shift`에 저장되어 있어 오류가 발생하므로, C언어와 다르게 `shift`를 0으로 초기화하였다.

배열 `d`에 '8, 5, 6, 2, 4, 1, 3, 7, 9, 10'과 같이 10개의 데이터가 저장되어 있다고 가정한다.

정렬할 자료는 10개지만 배열의 위치를 0부터 시작하기 때문에 `n`을 9로 초기화한다.

배열의 위치가 0부터 시작하므로 자료의 왼쪽 첫 번째 위치인 `left`를 0으로 초기화한다.

`left`가 `right`보다 작은 동안 ⑥~⑱번을 반복하여 수행한다.

수행 후 제어는 `while`문의 시작점인 ⑥번으로 이동하여 다시 시작한다.

배열의 위치가 0부터 시작하므로 `x`가 0에서 9까지 변하는 동안 배열의 값을 반복 출력한다.

033 삽입 정렬



선기 15.4, 06.7, 기사 07.7

문제

10개의 수치 자료를 입력 받아 배열에 저장한 후 저장된 자료를 오름차순으로 정렬하는 순서도를 작성하시오.

JAVA 코드

```
import java.util.Scanner;

public class Section033 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int j, i, k, key;
        ❷ int a[] = new int[10];
        ❸ j = -1; 배열의 위치가 0부터 시작하므로 배열 a의 첨자 j가 'j++(❸)'를 수행한 후 0이 되도록
            -1로 초기화한다.
        do do~while 반복문의 시작점으로 ❸~❹번을 반복하여 수행한다.
        {
            ❹ j++;
            ❺ a[j] = scanf.nextInt();
            ❻ } while (j < 9);
            for (i = 1; i <= 9; i++) 배열의 위치가 0부터 시작하므로 배열의 두 번째 값부터
                끝까지 비교하기 위해 i는 0이 아닌 1부터 9까지 반복 수
                행한다.
            {
                ❽ key = a[i];
                ❾ for (k = i - 1; k >= 0; k--) 배열의 위치가 0부터 시작하므로 배열의
                    첫 번째 위치는 1이 아니라 0이다. 그러므
                    로 k는 i-1부터 1까지가 아니라 0까지 변
                    하는 동안 반복 수행한다.
                {
                    ❿ if (a[k] > key)
                        a[k + 1] = a[k]; 수행 후 제어는 for문의 시작점인 ❸번으로 이
                            동하여 다시 시작한다.
                    else
                        break; for 반복문을 탈출한다. 제어가 ❶번으로 이동한다.
                }
                ㉑ a[k + 1] = key; 수행 후 제어는 for문의 시작점인 ❸번으로 이동하여 다시 시작
                    한다.
            }
            ㉒ for (i = 0; i <= 9; i++)
            ㉓ System.out.printf("%d ", a[i]);
            scanf.close();
        }
    }
}
```



IF문을 이용한 삽입 정렬

JAVA 코드

```

public class Section033_Wait {
    public static void main(String[] args) {
        ① int i, k, key;
        ① int a[] = { 9,7,4,6,3,1,8,2,5,10,19,17,14,16,13,11,18,
                    12,15,20 }; 배열 a에 20개의 데이터가 저장되어 있다고 가정한다.
        ② i = 1; 배열의 위치가 0부터 시작하므로 배열의 두 번째 값부터 비교하기 위해 i는 1로 초
            기화한다.
        do do~while 반복문의 시작점으로 ③~⑩번을 반복하여 수행한다.
        {
            ③ key = a[i];
            ④ k = i - 1;
            do do~while 반복문의 시작점으로 ⑤~⑧번을 반복하여 수행한다.
            {
                ⑤ if (key < a[k])
                {
                    ⑥ a[k + 1] = a[k];
                    ⑦ k--; 수행 후 제어는 ⑧번으로 이동한다.
                }
                else
                    break; do~while 반복문을 탈출한다. 제어는 ⑨번으로 이동한다.
            } while (k >= 0); 배열의 위치가 0부터 시작하므로 k가 0보다 크거나 같
                으면 회전 내에서 비교할 값이 남은 것이다.
            ⑨ a[k + 1] = key;
            ⑩ i++;
            ⑪ } while (i <= 19); 배열의 위치가 0부터 시작하므로 i가 19보다 작거나 같은 동
                안 KEY 값의 정렬을 수행한다.
        }
    }
}

```

034 석차 구하기



유형

1

배열 이용

선기 16.4, 11.4

문제

10명의 학생에 대한 중간고사 점수의 석차를 구하는 순서도를 작성하시오.

JAVA 코드

```

import java.util.Scanner;

public class Section034_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int m, n, i, j;
        ❷ int jumsu[] = new int[10];
        ❸ int rank[] = new int[10];
        ❹ m = -1; 배열의 위치가 0부터 시작하므로 배열 jumsu의 첨자 m이 'm++(❸)'를 수행한 후 0이 되
            도록 -1로 초기화한다.
        do ❶~while 반복문의 시작점으로 ❸~❹번을 반복하여 수행한다.
        {
            ❸ m++;
            ❹ jumsu[m] = scanf.nextInt();
            ❺ } while (m < 9);
            ❻ n = 9; 배열의 위치가 0부터 시작하므로 배열의 끝 위치로 사용될 n에 9를 저장한다.
            ❼ i = 0; 배열의 위치가 0부터 시작하므로 i를 0으로 초기화한다.
            ❽ while (i <= n) i가 n보다 작거나 같은 동안 ❹~❺번을 반복하여 수행한다.
            {
                ❾ rank[i] = 1;
                ❿ i++;
            }
            11 i = 0; 배열의 위치가 0부터 시작하므로 i를 0으로 초기화한다.
            12 while (i <= n) i가 n보다 작거나 같은 동안 14~16번을 반복하여 수행한다.
            {
                14 j = 0;
                15 while (j <= n) j가 n보다 작거나 같은 동안 17~19번을 반복하여 수행한다.
                {
                    17 if (jumsu[i] < jumsu[j])
                    18 rank[i]++;
                    19 j++;
                }
                16 i++; 수행 후 제어는 while문의 시작점으로 이동하여 12번부터 다시 시작한다.
            }
            13 for (int x = 0; x < 10; x++) jumsu 배열에 저장된 값을 모두 출력한다.
                System.out.printf("%d ", jumsu[x]);
                System.out.printf("\n");
                for (int x = 0; x < 10; x++) rank 배열에 저장된 값을 모두 출력한다.
                System.out.printf("%d ", rank[x]);
                scanf.close();
            }
        }
    }
}

```

선기 19.4, 16.4, 13.7, 11.4

문제

10명 학생의 국어, 수학 점수를 각각 입력 받아 총점을 계산한 후 총점 기준 내림차순으로 순위를 출력하는 순서도를 작성하시오. 단, 동점은 동석차로 하고 총점이 0인 경우는 출력하지 않는다.

JAVA 코드

```
import java.util.Scanner;

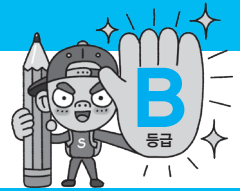
public class Section034_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);

        ❶ int i, j, r;
        ❶ int kuk[] = new int[10];
        ❶ int mat[] = new int[10];
        ❶ int hap[] = new int[10];
        ❷ i = -1; 배열의 위치가 0부터 시작하므로 배열의 첨자 i가 'i++(❸)'를 수행한 후 0이 되도록 -1로 초기화한다.

        do
        {
            ❸ i++;
            ❹ kuk[i] = scanf.nextInt();
            ❹ mat[i] = scanf.nextInt();
            ❺ hap[i] = kuk[i] + mat[i];
            ❻ } while (i < 9); 배열의 위치가 0부터 시작하므로 i가 9보다 작을 때까지 ❸~❺번을 반복 수행한다.

            ❼ for (i = 0; i <= 9; i++) 배열의 위치가 0부터 시작하므로 i는 0에서 9까지 반복 수행한다.
            {
                ❽ if (hap[i] != 0)
                {
                    ❾ r = 1;
                    ❿ for (j = 0; j <= 9; j++) 배열의 위치가 0부터 시작하므로 j는 0에서 9까지 반복 수행한다.
                    {
                        11 if (hap[i] < hap[j])
                        12 r++;
                    }
                    13 System.out.printf("%d %d %d %d %d\n", i+1, kuk[i], mat[i], hap[i], r);
                    0부터 시작하는 i를 번호로 출력하기 위해 +1을 해준다. 제어는 바깥쪽 for문의 시작점인 ❼번으로 이동하여 다시 시작한다.
                }
            }
        }
        scanf.close();
    }
}
```

035 이분 검색



유형

1

1차원 배열

문제

키보드로 입력 받은 값이 DATA 배열의 몇 번째에 기억되어 있는지 알아보기 위해 출력하는 순서도를 작성하시오. 단, DATA[10] 배열에는 10개의 숫자가 들어 있으며, 찾는 자료가 없을 경우 자료와 함께 "Not Found"를 출력하고 끝낸다.

JAVA 코드

```
import java.util.Scanner;

public class Section035_Type1 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int j, L, h, m;
        ① int data[] = { 8,15,35,55,60,61,70,80,92,99 }; data 배열에 10개의 숫자가 저장되어 있다고 가정한다.
        ② j = scanf.nextInt();
        ③ L = 0; 배열 위치가 0부터 시작하므로 검색 범위의 시작 위치인 L을 0으로 초기화한다.
        ④ h = 9; 배열 위치가 0부터 시작하므로 검색 범위의 마지막 위치는 10보다 1이 작은 9다. h를 9로 초기화한다.
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ⑤ if (L <= h)
            {
                ⑦ m = (L + h) / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.
                ⑧ if (j == data[m])
                {
                    ⑨ System.out.printf("%d %d", j, m + 1); 배열의 위치가 0부터 시작하므로 m이 0이면 첫 번째를 의미하므로 +1을 해준다.
                    break; while 반복문을 탈출한다. 제어가 ⑩번으로 이동한다.
                }
                ⑩ if (j < data[m])
                ⑪ h = m - 1; 수행 후 제어는 while문의 시작점인 ⑤번으로 이동하여 다시 시작한다.
                else
                ⑫ L = m + 1; 수행 후 제어는 while문의 시작점인 ⑤번으로 이동하여 다시 시작한다.
            }
            else
            {
                ⑥ System.out.printf("%d NOT FOUND", j);
                break; while 반복문을 탈출한다. 제어가 ⑬번으로 이동한다.
            }
        }
        ⑬ scanf.close();
    }
}
```

문제

키보드로 입력 받은 번호에 대한 점수를 DATA 배열에서 찾아 출력하는 순서도를 작성하시오. 단, DATA[2][10] 배열에는 번호와 점수가 들어 있다고 가정하고, 찾는 자료가 없을 경우 자료와 함께 “NOT FOUND”를 출력하고 끝낸다.

JAVA 코드

```
import java.util.Scanner;

public class Section035_Type2 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int j, L, h, m;
        ❶ int data[][] = { { 1,2,3,4,5,6,7,8,9,10 },
                        { 100,66,25,88,90,65,87,86,58,99 } };
                        data 배열에 2행 10열, 총 20개의 데이터가 저장되어 있다고 가정한다.
        ❷ j = scanf.nextInt();
        ❸ L = 0; 배열 위치가 0부터 시작하므로 검색 범위의 시작 위치인 L을 0으로 초기화한다.
        ❹ h = 9; 배열 위치가 0부터 시작하므로 검색 범위의 마지막 위치는 10보다 1이 작은 9다. h를 9로
            초기화한다.
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 { } 사이의 문장을 무한 반복한다.
        {
            ❺ if (L <= h)
            {
                ❷ m = (L + h) / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로
                    정수를 구하는 함수를 사용하지 않아도 된다.
                ❸ if (j == data[0][m]) 배열 위치가 0부터 시작하므로 1행의 값을 비교하
                    려면 행 첨자를 0으로 지정해야 한다.
                {
                    ❹ System.out.printf("%d %d\n", j, data[1][m]);
                        2행의 값을 출력하기 위해 행 첨자를 1로 지정한다.
                    break; while 반복문을 탈출한다. 제어가 ❸번으로 이동한다.
                }
                ❺ if (j < data[0][m])
                {
                    ❶ h = m - 1; 수행 후 제어는 while문의 시작점인 ❺번으로 이동하여 다시
                        시작한다.
                }
                ❻ else
                {
                    ❷ L = m + 1; 수행 후 제어는 while문의 시작점인 ❺번으로 이동하여 다시
                        시작한다.
                }
            }
            else
            {
                ❸ System.out.printf("%d NOT FOUND", j);
                break; while 반복문을 탈출한다. 제어가 ❸번으로 이동한다.
            }
        }
        ❶ scanf.close();
    }
}
```

036 병합(MERGE)



산기 18,6, 15,10, 10,7, 08,11

문제 다음의 처리 조건에 맞게 병합하는 순서도를 작성하시오.

〈처리 조건〉

- 1) 배열 A, B에는 정수가 오름차순으로 정렬되어 있다.
- 2) 데이터는 10건 미만이다.
- 3) 배열 A, B에서 0이 들어 있는 다음의 요소에는 데이터가 없는 것으로 간주한다.
- 4) 배열 A, B를 병합시켜 배열 C에 기억시키고 맨 마지막에는 0을 기억시킨다.
- 5) 배열 A와 B에 같은 데이터가 있으면 한 번만 C에 옮긴다.

예 배열 A의 자료와 배열 B의 자료를 배열 C에 병합한 결과

A[10] 1 3 5 6 0

B[10] 2 3 5 8 9 10 12 13 0

C[20] 1 2 3 5 6 8 9 10 12 13 0

JAVA 코드

```

public class Section036 {
    ❶ static int i, j, k; 정수형 전역 변수 i, j, k를 선언한다. 전역 변수이므로 메소드에서 변경
        되는 값이 전체에 영향을 미친다.

    public static void main(String[] args) {
        ❶ int a[] = { 2,3,5,8,9,10,12,13,0,0 }; 배열 a에 값이 저장되어 있다고
            가정한다. JAVA에서는 배열을 선언하면서 초기값을
            저장할 때 초기값의 개수만큼 배열의 크기가 정의되기
            때문에 10개의 요소를 갖도록 0값을 추가로 저장한다.

        ❶ int b[] = { 1,3,5,6,0,0,0,0,0,0 }; 배열 b에 값이 저장되어 있다고 가정한
            다. 위와 같은 이유로 0값을 추가로 저장한다.

        ❶ int c[] = new int[20];
        ❷ i = 0;
        ❸ j = 0;
        ❹ k = -1; 배열의 위치가 0부터 시작하므로 배열 첨자 k가 'k++(❺)'를 수행한 후 0이 되도록 -1로
            초기화한다.

        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 { } 사이의 문장을 무한 반복한다.

        {
            ❺ k++;
            ❻ if (a[i] < b[j])
                {
                    ❼ c[k] = a[i];
                    ❸ i++;
                    ❹ if (a[i] == 0)
                        ❶ BB(b, c); 배열 b와 c를 인수로 하여 BB 메소드를 호출한다. JAVA에서
                            배열은 객체이기 때문에 대괄호( ) 없이 b, c로 지정하면 b와
                            c 객체 안의 모든 요소를 인수로 하여 메소드를 호출한다.

                }

            ❸ else if (a[i] == b[j])
                {
                    ❶ c[k] = a[i];
                    ❷ i++;
                    ❸ j++;
                    ❹ if (a[i] == 0)
                        ❶ BB(b, c); 배열 b와 c를 인수로 하여 BB 메소드를 호출한다.
                    ❷ else if (b[j] == 0)
                        ❶ CC(a, c); 배열 a와 c를 인수로 하여 CC 메소드를 호출한다.
                }

            ❸ else
                {
                    ❶ c[k] = b[j];
                    ❷ j++;
                    ❸ if (b[j] == 0)
                        ❶ CC(a, c); 배열 a와 c를 인수로 하여 CC 메소드를 호출한다.
                }
        }
    }
}

```

```

22 static void BB(int b[], int c[]) { 클래스의 내부이기 때문에 static을, 메
    소드의 리턴 값이 없으므로 void를 붙인다.
    BB() 메소드의 인수로 정수 배열 b와 c를 받
    는다.

23 if (b[j] == 0)
    DD(c); b[j]가 0이면 배열 A의 자료를 모두 넣고 온 것이므로 병합이 끝난 것이다. 그러므
        로 DD 함수를 호출한다.
    do do~while 반복문의 시작점으로 24~26번을 반복하여 수행한다.
    {
24         k++;
25         c[k] = b[j];
26         j++;
27     } while (b[j] != 0); b[j]와 0이 같지 않은 동안 24~26번 문장을 반복하여 수행한다.
28     DD(c); 배열 c를 인수로 하여 DD 메소드를 호출한다.
    }

29 static void CC(int a[], int c[]) { 클래스의 내부이기 때문에 static을, 메
    소드의 리턴 값이 없으므로 void를 붙인다.
    CC() 메소드의 인수로 정수 배열 a와 c를 받
    는다.

    do do~while 반복문의 시작점으로 30~32번을 반복하여 수행한다.
    {
30         k++;
31         c[k] = a[i];
32         i++;
33     } while (a[i] != 0); a[i]와 0이 같지 않은 동안 30~32번 문장을 반복하여 수행한다.
34     DD(c); 배열 c를 인수로 하여 DD 메소드를 호출한다.
    }

35 static void DD(int c[]) { 클래스의 내부이기 때문에 static을, 메소드의 리턴 값이 없
    으므로 void를 붙인다. DD() 메소드의 인수로 정수 배열 c
    를 받는다.

36     k++;
37     c[k] = 0;
    System.exit(0); 프로그램을 종료한다.
    }
}

```

037 스택(STACK)



기사 18.10, 18.4

문제

10개의 요소를 갖는 1차원 배열을 스택으로 이용하는 순서도를 작성하시오. 입력되는 데이터가 스택의 크기보다 커지면 “Overflow”를 출력하고, 제거할 데이터가 없으면 “Underflow”를 출력한다.

JAVA 코드

```
import java.util.Scanner;

public class Section037 {
    static final int MAX = 5;  숫자 5를 MAX로 정의한다. 프로그램 안에서 MAX는 숫자 5와 동일하게 쓰인다.
    ① static int Stack[] = new int[MAX];  MAX개의 요소, 즉 5개의 요소를 갖는 정수형 배열 Stack을 전역 변수로 선언한다.
    ① static int Top = -1;  배열 Stack의 첨자 Top을 전역 변수로 선언하고 -1로 초기화한다. main() 메소드 밖에서 선언한 변수는 전역 변수로서 이 프로그램 전체에서 사용할 수 있으며, 값이 유지된다.
    ⑤ static int push(int ii) {  int 자료형을 리턴값으로 갖는 push 메소드의 시작점이다. ⑥번에서 push(k)라고 했으므로 정수형 변수 ii는 ⑥번에서 입력받은 k의 값을 받는다.
        ⑫ Top++;
        ⑬ if (Top >= MAX) {  배열의 첨자가 4에서 끝나므로 Top이 MAX의 값 5보다 크거나 같으면 ⑬~⑭번 문장을 실행하고, 아니면 ⑮번 문장으로 이동해서 실행을 계속한다.
            ⑮ System.out.printf("Overflow\n");
            ⑯ Top--;
            ⑰ return -1;  -1을 가지고 제어가 ⑮번으로 이동한다.
        }
        ⑱ Stack[Top] = ii;
        ⑲ return 0;  0을 가지고 제어가 ⑮번으로 이동한다.
    }

    ⑳ static int pop() {  int 자료형을 리턴값으로 갖는 pop 메소드의 시작점이다.
        ㉑ int r;
        ㉒ if (Top < 0) {  배열의 위치가 0부터 시작하므로 Top이 0보다 작으면 ㉒, ㉓번 문장을 실행하고, 아니면 ㉔번 문장으로 이동해서 실행을 계속한다.
            ㉓ System.out.printf("Underflow\n");
            ㉔ return -1;  -1을 가지고 제어가 ㉒번으로 이동한다.
        }
        ㉕ r = Stack[Top];
        ㉖ Top--;
        ㉗ return r;  제거된 값 r을 가지고 제어가 ㉖번으로 이동한다.
    }
}
```

```

public static void main(String[] args) {
    Scanner scanf = new Scanner(System.in);
    while (true) {   while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전
                    까지 { } 사이의 문장을 무한 반복한다.
2        int i, j, k;
        System.out.printf("작업을 선택하세요.:");
3        j = scanf.nextInt();   키보드로 정수를 입력받아 변수 j에 저장한다.
4        if (j == 1)
        {
            System.out.printf("삽입할 숫자를 입력하세요.:");
5            k = scanf.nextInt();   키보드로 정수를 입력받아 변수 k에 저장한다.
6            i = push(k);
7            if (i == -1) break;   while 반복문을 벗어난다. 제어가 ⑤번으로 이동
                                한다.
        }
11       else if (j == 2)
        {
12           i = pop();
13           if (i == -1) break;   while 반복문을 벗어난다. 제어가 ⑤번으로 이동
                                한다.
        }
        else
14         System.out.printf("제거한 자료는 %d입니다\n", i);
            실행 후 제어가 while 반복문의 시작점으로 이동한다.
        }
        else break;   while 반복문을 벗어난다. 제어가 ⑤번으로 이동한다.
    }
8    if (Top < 0)   배열의 위치가 0부터 시작하므로 0과 비교한다.
        System.out.printf("스택에 자료가 없습니다.\n");   실행 후 프로그램을
                                                            종료한다.
    else
    {
        System.out.printf("스택의 자료는 다음과 같습니다.\n");
9        for (int i = Top; i >= 0; i--)   배열의 위치가 0부터 시작하므로 배열
                                        Stack의 첨자 i는 Top에서 시작하
                                        여 1씩 감소하면서 0보다 크거나 같은
                                        동안 ⑩번을 반복하여 수행한다.
10        System.out.printf("%10d\n", Stack[i]);
    }
    scanf.close();
}
}

```



문제 4 다음은 원당 고등학교 2학년 7반 학생의 중간고사 점수를 파일로부터 입력 받아 총점을 계산한 후, 총점이 높은 순으로 석차를 부여하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.io.FileReader;  FileReader를 사용하기 위한 클래스이다.
import java.io.IOException;  파일 입출력 과정에서 발생하는 예외를 처리하기 위한 클래스이다.
import java.util.Scanner;

public class Test04 {
    public static void main(String[] args) throws IOException {
        입출력 과정에서 발생하는 예외 처리를 IOException 클래스로 넘긴다. JAVA에서는 파일 입출력 시
        예외 처리를 하지 않으면 오류가 발생한다.
        ①② Scanner inf = new Scanner(new FileReader("data01.
            txt")); Scanner 클래스의 객체 변수 inf를 생성하면서 FileReader를 사용해
            'data01.txt' 파일로부터 읽어오도록 만든다.
        ③ String name; Java에서는 문자열 변수를 사용할 수 있기 때문에, C언어에서와 같이
            문자 배열을 사용하지 않는다.
        ④ int kor, mat, eng, cnt, i, j;
        ⑤ String irum[] = new String[50];
        ⑥ int jumsu[] = new int[50];
        ⑥ int rank[] = new int[50];
        ⑦ cnt = -1; 배열의 위치가 0부터 시작하므로 배열의 첨자 cnt가 'cnt++(⑨)'를 수행한 후
            0이 되도록 -1로 초기화한다.
        ⑧ while(true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기
            전까지 ⑨~⑬번 문장을 무한 반복한다.
        {
        ⑨ cnt++; 자료의 수를 세기 위해 1씩 더한다.
        ⑩ if(!inf.hasNext()) 파일의 끝(EOF)을 파악하기 위해서 Scanner의 메소
            드 중 하나인 hasNext()를 사용한다. hasNext()
            는 다음 자료가 비어있을 때 False를 반환하기 때문
            에 부정 기호(!)를 넣는다.
        ⑪ break; while 반복문을 탈출한다. 제어가 ⑩번으로 이동한다.
        ⑩ name = inf.next(); 파일에서 문자열을 읽어 name에 저장한다.
        ⑩ kor = inf.nextInt(); 파일에서 숫자를 읽어 kor, mat, eng에 저장한다.
        ⑩ mat = inf.nextInt();
        ⑩ eng = inf.nextInt();
        ⑫ jumsu[cnt] = kor + mat + eng; 총점을 계산하여 jumsu[cnt]에
            저장한다.
        ⑬ irum[cnt] = name;
        }
```



```
14 cnt--; 입력된 자료의 실제 개수는 마지막 자료(EOF)를 제외시켜야 하므로 cnt에서 1을 차
    감한다.
15 i = 0; 배열의 위치가 0부터 시작하므로 석차의 위치를 지정할 i를 0으로 초기화한다.
16 while (i <= cnt) i가 cnt보다 작거나 같은 동안 17~18번을 반복 수행한다.
    {
17     rank[i] = 1; 다른 점수와 비교하기 전에는 1등이므로 rank[i]를 1로 초기화
        한다.
18     i++; 1증가시켜 다음 위치를 지정한다.
    }
19 i = 0; 배열의 위치가 0부터 시작하므로 석차의 위치를 지정할 i를 0으로 초기화한다.
20 while(i <= cnt) i가 cnt보다 작거나 같은 동안 21~26번을 반복 수행한다.
    {
21     j = 0; 배열의 위치가 0부터 시작하므로 비교 대상을 배열의 첫 번째로 지정하기 위
        해 0으로 초기화한다.
22     while (j <= cnt) j가 cnt보다 작거나 같은 동안 23~26번을 반복 수행한다.
        {
23         if (jumsu[i] < jumsu[j]) jumsu[i]가 jumsu[j]보다 작으면
            석차를 1증가시켜야 하므로 24번으
            로 가고, 아니면 25번으로 간다.
24             rank[i]++; 비교 기준 점수에 대한 석차 rank[i]를 1증가시킨다.
25             j++; j를 1증가시켜 다음 비교 대상 점수의 위치를 지정한다.
        }
26     i++; 비교 기준 점수가 있는 위치인 i를 0부터 cnt까지 변화시키기 위해 1씩 증가시
        킨다.
    }
27 for (int x = 0; x <= cnt; x++) x가 0에서 시작하여 cnt보다 작거나
    같은 동안 irum, jumsu, rank 배열
    에 들어 있는 모든 자료를 출력하고 끝
    낸다.
28     System.out.printf("%s %d %d\n", irum[x],
        jumsu[x], rank[x]);
29 inf.close(); 객체 변수 inf를 닫는다. 읽는 기능의 객체를 닫을 경우 메모리가 확보
    되고, 쓰는 기능의 객체를 닫을 경우 확실하게 저장하기 위함이다.
    }
}
```



문제 5 다음은 6학년 7반 학생의 중간고사 점수를 파일로부터 입력 받아 총점을 계산한 후, 특정 번호를 입력하면 그 학생의 총점을 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.io.FileReader;  FileReader를 사용하기 위한 클래스이다.
import java.io.IOException;  파일 입출력 과정에서 발생하는 예외를 처리하기 위한 클래스이다.
import java.util.Scanner;

public class Test05 {
    public static void main(String[] args) throws IOException {
        입출력 과정에서 발생하는 예외 처리를 IOException 클래스로 넘긴다. JAVA에서는 파일 입출력 시 예외 처리를 하지 않으면 오류가 발생한다.
        ①② Scanner inf = new Scanner(new FileReader("data02.txt"));
        Scanner 클래스의 객체 변수 inf를 생성하면서 FileReader를 사용해 'data02.txt' 파일로부터 읽어오도록 만든다.
        Scanner scanf = new Scanner(System.in);
        ③ int cnt, j, L, h, m, bun, kor, eng, mat;
        ④ int data[][] = new int[2][30];
        ⑤ cnt = -1;  배열의 위치가 0부터 시작하므로 배열의 첨자 cnt가 'cnt++(⑦)'을 수행한 후 0이 되도록 -1로 초기화한다.
        ⑥ while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ⑦~⑪번 문장을 무한 반복한다.
        {
            ⑦ cnt++;  자료의 수를 세기 위해 1씩 더한다.
            ⑧ if(!inf.hasNext())  파일의 끝(EOF)을 파악하기 위해서 Scanner의 메소드 중 하나인 hasNext()를 사용한다. hasNext()는 다음 자료가 비어있을 때 False를 반환하기 때문에 부정 기호(!)를 넣는다.
            ⑨ break;  while 반복문을 탈출한다. 제어가 ⑫번으로 이동한다.
            ⑧ bun = inf.nextInt();  파일에서 정수를 읽어서 bun, kor, eng, mat에 저장한다.
            ⑧ kor = inf.nextInt();
            ⑧ eng = inf.nextInt();
            ⑧ mat = inf.nextInt();
            ⑩ data[0][cnt] = bun;  번호를 저장한다.
            ⑪ data[1][cnt] = kor + mat + eng;  총점을 계산하여 저장한다.
        }
    }
}
```



```
12 cnt--; 입력된 자료의 실제 개수는 마지막 자료(EOF)를 제외시켜야 하므로 cnt에서 1을 차감한다.
13 j = scanf.nextInt(); 찾을 번호 j를 입력받는다.
14 L = 0; 배열의 위치가 0부터 시작하므로 검색 범위의 시작 위치인 L을 0으로 초기화한다.
15 h = cnt; 검색 범위의 마지막 위치는 입력 받은 데이터의 개수 cnt이므로 h를 cnt로 초기화한다.
16 while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 17
    ~28번을 무한 반복한다.
    {
17     if (L <= h) L이 h보다 작거나 같으면 자료를 찾기 위해 18번으로 가고, 아니면 시작 위치
        가 마지막 위치보다 커진 것이므로 20번으로 간다.
        {
18         m = (L + h) / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수
            를 구하는 함수를 사용하지 않아도 된다.
19         if (j == data[0][m]) 찾을 번호 j와 범위의 중간 위치에 있는 값 data[0][m]
            이 같으면 값을 찾은 것이므로 20번으로 가고, 아니면
            22번으로 간다.
            {
20                 System.out.printf("%d %d", j, data[1][m]);
                    입력 받은 j와 찾은 위치에 있는 점수 data[1][m]을 출력한다.
21                 break; while 반복문을 탈출한다. 제어가 20번으로 이동한다.
            }
22         if (j < data[0][m]) 찾을 번호 j가 data[0][m]보다 작으면 중간 위치(m) 이
            전 범위에 찾는 번호가 있다는 것이므로 23번으로 가고,
            아니면 중간 위치(m) 이후 범위에 찾는 번호가 있다는 것
            이므로 25번으로 간다.
23                 h = m - 1; 검색 범위의 마지막 위치 h를 중간 위치보다 1칸 이전으로 변경하
                    고 23번으로 간다.
24         else
25                 L = m + 1; 검색 범위의 마지막 위치 h를 중간 위치보다 1칸 이후로 변경하고
                    25번으로 간다.
        }
26     else
    {
27         System.out.printf("%d NOT FOUND", j); 자료를 찾지 못한 경
            우다. 번호와 'NOT
            FOUND'를 출력한다.
28         break; while 반복문을 탈출한다. 제어가 20번으로 이동한다.
    }
29 }
inf.close(); 객체 변수 inf와 scanf를 닫는다. 읽는 기능의 객체를 닫을 경우 메모리가 확보되
    고, 쓰는 기능의 객체를 닫을 경우 확실하게 저장하기 위함이다.
scanf.close();
}
```



문제 6 다음은 데이터가 정렬되어 저장되어 있는 배열 `a[L]`과 배열 `b[m]`을 병합(Merge)하여 새로운 배열 `c[n]`에 순서대로 저장하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test06 {
    public static void main(String[] args) {
        ① int t, j, k, p;
           int L = 5, m = 9, n = 20;  L, m, n은 배열 a, b, c의 크기로, 각각 5, 9, 20으로 가정한다.

           int a[] = { 1,3,5,6,7 };  배열 a에 5개 데이터가 저장되어 있다고 가정한다.
           int b[] = { 2,3,5,8,9,10,12,13,14 };  배열 b에 9개의 데이터가 저장되어 있다고 가정한다.

           int c[] = new int[n];  배열 선언 시 변수 n을 사용하였다. JAVA에서는 사용하지 않는 변수가 있을 경우 주의 메시지가 표시된다.

        ② t = j = k = 0;  배열의 위치는 0부터 시작하므로 첫 번째 자료를 지정하기 위한 첨가 변수들을 0으로 초기화한다.

        ③ while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ④~⑪번을 무한 반복한다.

        {
            ④ if (a[t] < b[j])  a[t]가 b[j]보다 작으면 배열 a의 자료를 배열 c에 넣기 위해 ⑤번으로 가고, 아니면 배열 b의 자료를 배열 c에 넣기 위해 ⑩번으로 간다.

                {
                    ⑤ c[k] = a[t];  배열 a의 자료를 배열 c에 저장한다.
                    ⑥ t++;  배열 a의 위치를 다음 위치로 옮기기 위해 t를 1씩 증가시킨다.
                    ⑦ k++;  배열 c의 위치를 다음 위치로 옮기기 위해 k를 1씩 증가시킨다.
                    ⑧ if (t > L - 1)  배열의 위치가 0부터 시작하므로 배열 a의 크기는 'L-1'이다. t가 배열의 크기보다 크면 배열 a의 자료를 모두 처리한 것이므로 ⑨번으로 가고, 아니면 병합할 자료가 남은 것이므로 ④번으로 간다.

                        {
                            ⑨ for (p = j; p <= m-1; p++)  배열의 위치가 0부터 시작하므로 배열 b의 크기는 'm-1'이다. 이곳으로 온 경우는 배열 a의 자료를 모두 처리한 경우이므로 이후에는 배열 b의 자료만을 배열 c에 넣기 위해 배열 a의 자료가 모두 처리될 당시의 배열 b의 위치부터 b 배열의 크기인 'm-1'까지 배열 위치를 1씩 증가시키면서 ⑩~⑪번을 반복 수행한다.

                                {
                                    ⑩ c[k] = b[p];  배열 b의 자료를 배열 c에 저장한다.
                                    ⑪ k++;  배열 c의 위치를 다음 위치로 옮기기 위해 k를 1씩 증가시킨다.

                                }

                            }

                        }

                }
        }
    }
}
```



```
12         break; while 반복문을 탈출한다. 제어가 22번으로 이동한다.
        }
13     }
    else
    {
14         c[k] = b[j]; 배열 b의 자료를 배열 c에 저장한다.
15         j++; 배열 b의 위치를 다음 위치로 옮기기 위해 j를 1씩 증가시킨다.
16         k++; 배열 c의 위치를 다음 위치로 옮기기 위해 k를 1씩 증가시킨다.
17         if (m-1 < j) 배열의 위치가 0부터 시작하므로 배열 b의 크기는 'm-1'이다. j가 배열의 크기보다 크면 배열 b의 자료를 모두 처리한 것이므로 18번으로 가고, 아니면 병합할 자료가 남은 것이므로 14번으로 간다.

        {
18             for (p = t; p <= L-1; p++) 배열의 위치가 0부터 시작하므로 배열 a의 크기는 'L-1'이다. 이곳으로 온 경우는 배열 b의 자료를 모두 처리한 경우이므로 이후에는 배열 a의 자료만을 배열 c에 넣기 위해 배열 b의 자료가 모두 처리될 당시의 배열 a의 위치부터 a 배열의 크기인 'L-1'까지 배열 위치를 1씩 증가시키면서 19~20번을 반복 수행한다.

            {
19                 c[k] = a[p]; 배열 a의 자료를 배열 c에 저장한다.
20                 k++; 배열 c의 위치를 다음 위치로 옮기기 위해 k를 1씩 증가시킨다.
            }
21         break; while 반복문을 탈출한다. 제어가 22번으로 이동한다.
        }
    }
}
22 }
```



산기 18년 4월

문제 7 다음은 배열에 저장된 값을 오름차순으로 정렬하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

public class Test {
    ④ public static int[] arr(int[] a) {
        ⑤ int i, j, sw, temp, n = 4;
        ⑥ if(a[0] == 0 || a[0] < 1)
        ⑦ return a;
        ⑧ for(i = 0; i < n-1; i++) {
            ⑨ sw = i;
            ⑩ for(j = i + 1; j < n; j++) {
                ⑪ if(a[j] < a[sw])
                ⑫ sw = j;
            }
            ⑬ temp = a[i];
            ⑭ a[i] = a[sw];

```

③에서 'arr(n)'라고 했으므로 a는 n 배열의 시작 주소를 받아 다음과 같이 사용될 수 있다.

a[0]	a[1]	a[2]	a[3]
4	2	7	1

n[0] n[1] n[2] n[3]

정수형 변수 i, j, sw, temp, n을 선언하고 n을 4로 초기화한다.

a[0]의 값이 0이거나 1보다 작으면 ⑦번으로 가고 아니면 ⑨번 문장을 실행한다.

a를 반환한다. 즉 배열 a의 시작 주소를 가지고 arr 함수를 호출했던 main() 함수로 제어를 옮긴다.

반복 변수 i가 0에서 시작하여 1씩 증가하면서 n-1보다 작은 동안 ⑨~⑫번을 반복하여 수행한다. 결국 ⑨~⑫번을 4번 반복하여 수행한다.

i의 값을 sw에 저장한다.

반복 변수 j가 i+1에서 시작하여 1씩 증가하면서 n보다 작은 동안 ⑪~⑫번을 반복하여 수행한다.

- i가 0일 때 j는 1에서 3까지 3회 반복
- i가 1일 때 j는 2에서 3까지 2회 반복
- i가 2일 때 j는 3에서 3까지 1회 반복
- i가 3일 때 j는 n과 같아져 수행하지 않으므로 총 6회 수행된다.

a[j]의 값이 a[sw]의 값보다 작으면 ⑫번을 수행하고 아니면 ⑩번부터 다시 시작한다.

j의 값을 sw에 저장한다.

a[i]의 값과 a[sw]의 값을 서로 교환한다.



```
15         a[sw] = temp;
16     }
17     return a;
18 }

public static void main(String[] args) {
19     int i;           정수형 변수 i를 선언한다.
20     int n[] = {4,2,7,1}; 배열을 선언할 때 사용할 개수를 생략하고 초기값을 지정하면, 초기값
                        으로 지정된 값의 수와 같은 크기의 배열이 선언된다.

                        n[0] n[1] n[2] n[3]
배열 n   4   2   7   1
21     arr(n); 인수로 배열의 이름을 지정하면 배열의 시작 주소가 인수로 전달된다. 그러므로 arr(n)은
                        arr(&n[0])과 같은 의미이다. arr(n)을 호출한다.
22     for(i = 0; i < 4; i++) 반복 변수 i가 0에서 시작하여 1씩 증가하면서 4보다 작
                        은 동안 23번을 반복하여 수행한다. 결국 23번을 총 4번
                        반복한다.
23         System.out.printf(n[i]+" "); n[i]의 값과 공백 2칸(" ")을 함께 출력한다.
                        결과 1 2 4 7
24     }
25 }
```

038 배열 1 - 기본 5행 5열



유형

1

행 고정 열 변화

문제 5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

JAVA 코드

```

public class Section038_Type1 {
    public static void main(String[] args) {
        ❶ int i, j, k;
        ❶ int a[][] = new int[5][5];
        ❷ k = 0;
        ❸ for (i = 0; i <= 4; i++)    배열의 위치를 0부터 시작하기 때문에 행 첨자 i는 0~4
                                   까지 변하는 동안 ❹~❺번을 반복 수행한다.
        {
            ❹ for (j = 0; j <= 4; j++)    배열의 위치가 0부터 시작하기 때문에 열 첨자 j는
                                   0~4까지 변하는 동안 ❻~❼번을 반복 수행한다.
            {
                ❺ k++;
                ❻ a[i][j] = k;
            }
        }
        ❼ for (int x = 0; x <= 4; x++)    5행을 모두 출력하기 위해 행 첨자 x는 0~4까지
                                   5회 반복 수행한다.
        {
            for (int y = 0; y <= 4; y++)    각 행의 5열을 모두 출력하기 위해 열 첨
                                   자 y는 0~4까지 5회 반복 수행한다.
                System.out.printf("%3d", a[x][y]);    a[x][y]의 값을 3칸의
                                                         정수로 출력한다.
            System.out.printf("\n");    다음 행을 출력하기 위해 커서를 다음 줄의 처음
                                   으로 옮긴다.
        }
    }
}

```

유형

2

열 고정 행 변화

문제

5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

JAVA 코드

```

public class Section038_Type2 {
    static final int LENGTH = 5;  클래스 안에서 LENGTH는 항상 고정된 값 5를 갖도록 정의
                                  한다.
    public static void main(String[] args) {
        ①    int i, j, k;
        ①    int a[][] = new int[5][5];
        ②    k = 0;
        ③    for (i = 0; i <= LENGTH-1; i++)  배열의 위치는 0부터 시작한다. LENGTH가
                                              5로 정의되었으므로 행 첨자 i는 0~4까지
                                              변하는 동안 ④~⑥번을 반복 수행한다.

        {
            ④    for (j = 0; j <= LENGTH-1; j++)  LENGTH가 5이므로 열 첨자 j는
                                              0~4까지 변하는 동안 ⑤~⑥번을
                                              반복 수행한다.

            {
                ⑤    k++;
                ⑥    a[j][i] = k;
            }
        }
        ⑦    for (int x = 0; x <= LENGTH-1; x++)  5행을 모두 출력하기 위해 행 첨자 x
                                              는 0~4까지 5회 반복 수행한다.

        {
            for (int y = 0; y <= LENGTH-1; y++)  각 행의 5열을 모두 출력하기
                                              위해 열 첨자 y는 0~4까지 5
                                              회 반복 수행한다.

            System.out.printf("%3d", a[x][y]);  a[x][y]의 값을 3칸의
                                              정수로 출력한다.

            System.out.printf("\n");  다음 행을 출력하기 위해 커서를 다음 줄의 처음
                                              으로 옮긴다.

        }
    }
}

```

039 배열 2 - 직각 삼각형 만들기



유형

1

왼쪽 직각 삼각형 만들기

문제

5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1				
2	3			
4	5	6		
7	8	9	10	
11	12	13	14	15

JAVA 코드

```

public class Section039_Type1 {
    public static void main(String[] args) {
        ① int i, j, k;
        ① int a[][] = new int[5][5];  JAVA에서 객체를 선언할 때 특별한 값을 저장하지
                                   않으면 정수형인 경우 0으로 초기화된다. 배열 또한
                                   객체이므로 따로 0으로 초기화하는 과정이 필요 없다.

        ② k = 0;
        ③ for (i = 0; i <= 4; i++)  배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4
                                   까지 변하는 동안 ④~⑥번을 반복 수행한다.
        {
            ④ for (j = 0; j <= i; j++)  배열의 위치가 0부터 시작하기 때문에 열 첨자 j
                                       는 0에서 시작해서 해당 회전 수인 i까지 변하는
                                       동안 ⑤~⑥번을 반복 수행한다.
            {
                ⑤ k++;
                ⑥ a[i][j] = k;
            }
        }
        ⑦ for (int x = 0; x <= 4; x++)  5행을 모두 출력하기 위해 행 첨자 x는 0~4까지
                                       5회 반복 수행한다.
        {
            for (int y = 0; y <= 4; y++)  각 행의 5열을 모두 출력하기 위해 열 첨
                                       자 y는 0~4까지 5회 반복 수행한다.
                System.out.printf("%3d", a[x][y]);  a[x][y]의 값을 3칸의
                                                       정수로 출력한다.
            System.out.printf("\n");  다음 행을 출력하기 위해 커서를 다음 줄의 처음
                                       으로 옮긴다.
        }
    }
}

```

유형

2

오른쪽 직각 삼각형 만들기

산기 19.10

문제

5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	2	3	4	5
	6	7	8	9
		10	11	12
			13	14
				15

JAVA 코드

```

public class Section_Type2 {
    public static void main(String[] args) {
        ① int i, j, k;
        ① int a[][] = new int[5][5];  JAVA에서 객체를 선언할 때 특별한 값을 저장하지
                                   않으면 정수형인 경우 0으로 초기화된다. 배열 또한
                                   객체이므로 따로 0으로 초기화하는 과정이 필요 없다.

        ② k = 0;
        ③ for (i = 0; i <= 4; i++)  배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4
                                   까지 변하는 동안 ④~⑥번을 반복 수행한다.
        {
            ④ for (j = i; j <= 4; j++)  배열의 위치가 0부터 시작하기 때문에 열 첨자 j
                                       는 '행 번호'에서 시작해서 4까지 변하는 동안 ⑤
                                       ~⑥번을 반복 수행한다.
            {
                ⑤ k++;
                ⑥ a[i][j] = k;
            }
        }
        ⑦ for (int x = 0; x <= 4; x++) 5행을 모두 출력하기 위해 행 첨자 x는 0~4까지
                                       5회 반복 수행한다.
        {
            for (int y = 0; y <= 4; y++)  각 행의 5열을 모두 출력하기 위해 열 첨
                                       자 y는 0~4까지 5회 반복 수행한다.
                System.out.printf("%3d", a[x][y]);  a[x][y]의 값을 3칸의
                                                         정수로 출력한다.
            System.out.printf("\n");  다음 행을 출력하기 위해 커서를 다음 줄의 처음
                                       으로 옮긴다.
        }
    }
}

```

문제 5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1				
3	2			
6	5	4		
10	9	8	7	
15	14	13	12	11

JAVA 코드

```

public class Section039_Type3 {
    public static void main(String[] args) {
        ❶ int i, j, k;
        ❶ int a[][] = new int[5][5];  JAVA에서 객체를 선언할 때 특별한 값을 저장하지
                                   않으면 정수형인 경우 0으로 초기화된다. 배열 또한
                                   객체이므로 따로 0으로 초기화하는 과정이 필요 없다.

        ❷ k = 0;
        ❸ for (i = 0; i <= 4; i++)  배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4
                                   까지 변하는 동안 ❹~❺번을 반복 수행한다.

        {
            ❹ for (j = i; j >= 0; j--)  배열의 위치가 0부터 시작하기 때문에 열 첨자 j
                                   는 열의 해당 회전 수를 나타내는 행 번호에서 시
                                   작해서 0까지 변하는 동안 ❺~❻번을 반복 수행
                                   한다.

                {
                    ❺ k++;
                    ❻ a[i][j] = k;
                }
        }

        ❽ for (int x = 0; x <= 4; x++)  5행을 모두 출력하기 위해 행 첨자 x는 0~4까지
                                   5회 반복 수행한다.

        {
            for (int y = 0; y <= 4; y++)  각 행의 5열을 모두 출력하기 위해 열 첨
                                   자 y는 0~4까지 5회 반복 수행한다.
                System.out.printf("%3d", a[x][y]);  a[x][y]의 값을 3칸의
                                                       정수로 출력한다.

                System.out.printf("\n");  다음 행을 출력하기 위해 커서를 다음 줄의 처음
                                   으로 옮긴다.
            }
        }
    }
}
    
```

040 배열 3 - 'ㄹ' 자로 채우기



산기 09.4, 기사 18.6, 13.10, 11.7

문제 5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16
21	22	23	24	25

JAVA 코드

```

public class Section040 {
    public static void main(String[] args) {
        ① int k, L, m, n, i, j, p;
        ① int a[][] = new int[5][5];
        ② k = 0;
        ③ L = 0; 배열의 위치가 0부터 시작하기 때문에 첫 행의 열 시작 위치를 0으로 지정하기 위해 L을 0으로 초기화한다.
        ④ m = 4; 배열의 위치가 0부터 시작하기 때문에 첫 행의 열 끝 위치를 4로 지정하기 위해 m을 4로 초기화한다.
        ⑤ n = 1;
        ⑥ for (i = 0; i <= 4; i++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4까지 변하는 동안 ⑦~⑧번을 반복 수행한다.
        {
            ⑦ for (j = L; j != m + n; j += n) 열 첨자 j는 L부터 시작하여 j와 (m+n)이 같지 않을 때까지 반복 수행한다.
                                   결국 j는 한 번은 0~4까지 5회 반복, 또 한 번은 4~0까지 5회 반복을 번갈아
                                   가면서 수행한다.
            {
                ⑧ k++;
                ⑨ a[i][j] = k;
            }
            ⑩ p = L;
            ⑪ L = m;
            ⑫ m = p;
            ⑬ n *= -1; 'n = n * -1'과 동일하다. 열 위치를 한 번은 1씩 증가시키고 또 한 번은 1씩 감소시키기 위해 n에 -1을 곱한다.
        }
        ⑭ for (int x = 0; x <= 4; x++)
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");
        }
    }
}

```



기사 09.4

문제

5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

		1		
	2	3	4	
5	6	7	8	9
	10	11	12	
		13		

JAVA 코드

```

public class Section041 {
    public static void main(String[] args) {
        ❶ int i, j;
        ❶ int a[][] = new int[5][5];
        ❷ int k = 0;
        ❸ int s = 2, e = 2; 배열의 위치를 0부터 시작하기 때문에 0행에서의 열의 시작과 끝은 2
                               이로 열의 시작 s와 열의 끝 e를 2로 초기화한다.
        ❹ for (i = 0; i <= 4; i++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4
                               까지 변하는 동안 ❹~❷번을 반복 수행한다.
        {
            ❺ for (j = s; j <= e; j++) 열 첨자는 열의 시작인 s에서 시작해서 열의 끝인
                               e까지 변하는 동안 ❺~❷번을 반복 수행한다.
            {
                ❻ k++;
                ❼ a[i][j] = k;
            }
            ❽ if (i >= 2) 행의 중간 위치인지를 판단한다. 배열의 위치를 0부터 시작하기 때문에 0
                               에서 4까지 증가하는 행의 중간 위치는 2가 된다.
            {
                ❾ s++;
                ❿ e--; 수행 후 제어는 for문의 시작점인 ❹번으로 이동하여 다시 시작한다.
            }
            else
            {
                11 s--;
                12 e++; 수행 후 제어는 for문의 시작점인 ❹번으로 이동하여 다시 시작한다.
            }
        }
        13 for (int x = 0; x <= 4; x++)
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");
        }
    }
}

```

042 배열 5 - 모래시계 만들기



기사 15,7, 14,11, 12,7, 10,10

문제

배열의 크기를 입력 받아 배열의 크기에 따라 아래와 같이 기억시키는 순서도를 작성하시오. 단, 배열의 크기는 홀수로만 입력이 가능하고, 여기에서는 5를 입력 받았다고 가정한다.

1	2	3	4	5
	6	7	8	
		9		
	10	11	12	
13	14	15	16	17

JAVA 코드

```

import java.util.Scanner;

public class Section042 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int x, m, L, i, j, k;
        ② x = scanf.nextInt();
        ③ int a[][] = new int[x][x];
        ④ k = 0;
        ⑤ L = x; 배열의 위치가 0부터 시작하기 때문에 첫 행에서는 입력 받은 배열의 '크기-1' 만큼 숫자를
            저장하는데, 'L---(⑤)'를 수행한 후 숫자를 저장하므로 L을 x로 초기화한다.
        ⑥ m = x / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수를 사
            용하지 않아도 된다. 배열의 위치가 0부터 시작하기 때문에 0에서 x까지 증가하는
            행의 중간 위치는 x를 2로만 나누면 된다.
        ⑦ for (i = 0; i <= m; i++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0부
            터 행의 중간 위치인 m까지 변하는 동안 ⑧~⑪번을 반복
            수행한다.

        {
            ⑧ L--;
            ⑨ for (j = i; j <= L; j++) 열 첨자 j는 행 첨자 i에서 시작해서 1씩 줄어드
                는 L까지 변하는 동안 ⑩~⑪번을 반복 수행한다.

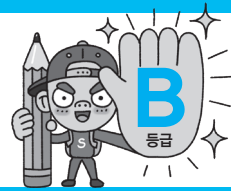
            {
                ⑩ k++;
                ⑪ a[i][j] = k;
            }
        }
        ⑫ for (i = m + 1; i <= x - 1; i++) 행 첨자 i는 중간 행 다음부터 시작하므로
            m+1에서 시작해서 마지막 행인 x-1(배열
            의 위치가 0부터 시작하기 때문에)까지 반
            복 수행한다.

        {
            ⑬ L--;
            ⑭ for (j = L; j <= i; j++) 열 첨자 j는 1씩 줄어드는 L부터 행 번호인 i까지
                반복하여 수행한다.

            {
                ⑮ k++;
                ⑯ a[i][j] = k;
            }
        }
        ⑰ for (int y = 0; y < x; y++)
        {
            ⑱ for (int z = 0; z < x; z++)
            ⑲ System.out.printf("%3d", a[y][z]);
            ⑳ System.out.printf("\n");
        }
        ㉑ scanf.close();
    }
}

```

043

배열 6 - 오른쪽에 빈
삼각형 만들기

문제

배열의 크기를 입력 받아 배열의 크기에 따라 아래와 같이 기억시키는 순서도를 작성하시오. 단, 배열의 크기는 홀수로만 입력이 가능하고 여기서는 배열의 크기로 7을 입력 받았다고 가정한다.

1	2	3	4	5	6	7
8	9	10	11	12	13	
14	15	16	17	18		
19	20	21	22			
23	24	25	26	27		
28	29	30	31	32	33	
34	35	36	37	38	39	40

JAVA 코드

```

import java.util.Scanner;

public class Section043 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int x, m, k, i, j, L;
        ② x = scanf.nextInt();
        ③ int a[][] = new int[x][x];
        ④ k = 0;
        ⑤ m = x / 2; 정수형 변수는 소수점 이하를 버리고 정수만 저장하므로 정수를 구하는 함수를 사용
            하지 않아도 된다. 배열의 위치가 0부터 시작하기 때문에 0에서 x까지 증가하는 행
            의 중간 위치는 x를 2로만 나누면 된다.
        ⑥ for (i = 0; i <= x - 1; i++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 i
            는 0부터 x까지 변하는 동안 ⑦~⑫번을 반복 수행
            한다.

        {
            ⑦ if (i < m)
            ⑧     L = x - i; 중간 행 이전에 사용할 열의 끝 위치는 (입력된 크기+1)에서 행의 증
                가한 행 수만큼 줄어들지만 배열의 위치가 0부터 시작하기 때문에
                (x+1)-i이 아닌 x-i가 된다.

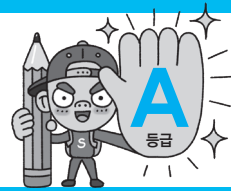
                else
            ⑨     L = i + 1; 중간 행부터 사용할 열의 끝 위치는 회전수를 나타내는 행 번호와 같
                지만 배열의 위치가 0부터 시작하기 때문에 i가 아닌 i+1이 된다.
            ⑩ for (j = 0; j <= L - 1; j++) 배열의 위치가 0부터 시작하기 때문에 열
                첨자 j는 0부터 L-1까지 변하는 동안 ⑪~
                ⑫번을 반복 수행한다.

                {
                    ⑪ k++;
                    ⑫ a[i][j] = k;
                }
        }
        ⑬ for (int y = 0; y < x; y++)
        {
            for (int z = 0; z < x; z++)
                System.out.printf("%3d", a[y][z]);
            System.out.printf("\n");
        }
        scanf.close();
    }
}

```

044

배열 7 - 이등변 삼각형 만들기



기사 18,4, 15,7, 14,11, 12,7, 10,10

문제

배열의 열 크기를 입력 받아 배열의 크기에 따라 아래와 같이 기억시키는 순서도를 작성하시오. 단, 여기서는 배열의 열 크기로 4를 입력 받았다고 가정한다.

			10
		5	11
	2	6	12
1	3	7	13
	4	8	14
		9	15
			16

JAVA 코드

```

import java.util.Scanner;

public class Section044 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int col, row, k, j, L, e, i;
        ❷ col = scanf.nextInt();
        ❸ row = col * 2 - 1;
        ❹ int a[][] = new int[row][col];
        ❺ k = 0;
        ❻ for (j = 0; j <= col - 1; j++)
            {
                ❼ L = col - (j + 1);
                ❽ e = (j - 1) + col;
                ❾ for (i = L; i <= e; i++)
                {
                    ❿ k++;
                    ⓫ a[i][j] = k;
                }
            }
        ⓬ for (int y = 0; y < row; y++)
        {
            for (int z = 0; z < col; z++)
                System.out.printf("%3d", a[y][z]);
            System.out.printf("\n");
        }
        scanf.close();
    }
}

```

배열의 위치가 0부터 시작하기 때문에 열 첨
자 j는 0부터 (col-1)까지 변하는 동안 ❷~
⓫번을 반복 수행한다.

열에서 행의 시작 위치는 열이 증가할 때마다 증가한 열의 수
만큼 줄어들어 'col-j'가 되어야 하지만 배열의 위치가 0부
터 시작하기 때문에 'col-(j+1)'이 된다.

행의 끝 위치는 회전수를 나타내는 열 번호에 일정한 크기(열
의 크기보다 1작은 크기)가 더해져야 하지만 배열의 위치가 0
부터 시작하기 때문에 회전수를 나타내는 (열 번호-1)에 열의
크기가 더해진 '(j-1)+col'이 된다.

045 배열 8 - 90도 회전하기



신기 11, 11, 기사 14, 7, 10, 7

문제

배열에 A와 같이 입력한 후 90도 회전한 결과를 배열 B에 이동하는 순서도를 작성하시오.

배열 A

11	12	13	14	15
	7	8	9	10
		4	5	6
			2	3
				1

배열 B

				11
			7	12
		4	8	13
	2	5	9	14
1	3	6	10	15

JAVA 코드

```

public class Section045 {
    public static void main(String[] args) {
        ①      int i, j, k;
        ①      int a[][] = new int[5][5];
        ①      int b[][] = new int[5][5];
        ②      k = 0;
        ③      for (i = 4; i >= 0; i--)   배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 4~0
                                         까지 변하는 동안 ④~⑥번을 반복 수행한다.
        {
            ④      for (j = i; j <= 4; j++)   배열의 위치가 0부터 시작하기 때문에 열 첨자 j
                                                는 i에서 시작해서 4까지 변하는 동안 ⑤~⑥번
                                                을 반복 수행한다.
            {
                ⑤      k++;
                ⑥      a[i][j] = k;
            }
        }
        ⑦      for (i = 0; i <= 4; i++)   배열의 위치가 0부터 시작하기 때문에 행 첨자 i는 0~4
                                         까지 변하는 동안 ⑧~⑨번을 반복 수행한다.
        {
            ⑧      for (j = 0; j <= 4; j++)   배열의 위치가 0부터 시작하기 때문에 열 첨자 j
                                                는 0~4까지 변하는 동안 ⑨번을 반복 수행한다.
            ⑨      b[j][4 - i] = a[i][j];   B 배열에서는 j를 행으로 사용한다. 배열의
                                                위치가 0부터 시작하기 때문에 열은 4~0까지
                                                변해야 하므로 (4-i)가 된다.
        }
        ⑩      for (int x = 0; x <= 4; x++)   A 배열 출력
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");   한 줄 띄운다.
        }
        System.out.printf("\n");   한 줄 띄운다.
        for (int x = 0; x <= 4; x++)   B 배열 출력
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", b[x][y]);
            System.out.printf("\n");   한 줄 띄운다.
        }
    }
}

```

046 배열 9 - 달팽이 만들기



신기 19,10, 14,4, 11,7, 기사 18,4, 06,7

문제

5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

JAVA 코드

```

public class Section046 {
    public static void main(String[] args) {
        ①      int k, c, i, j, f, n;
        ①      int a[][] = new int[5][5];
        ②      k = 0;
        ③      c = 1;
        ④      i = 0;   배열의 위치가 0부터 시작하므로 행 위치를 1부터 지정하기 위해 i를 0으로 초기화한다.
        ⑤      j = -1;   배열의 위치가 0부터 시작하므로 열 위치를 나타내는 j가 'j += c(⑨)'를 수행한 후 0이
                        되도록 -1로 초기화한다.
        ⑥      f = 5;
        while(true)   while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지
                        { } 사이의 문장을 무한 반복한다.
        {
            ⑦      for (n = 1; n <= f; n++)
            {
                ⑧      k++;
                ⑨      j += c;
                ⑩      a[i][j] = k;
            }
            ⑪      f--;
            ⑫      if (f <= 0)   f, 즉 각 회전의 수행 횟수가 5~10이 될 때까지 while문(⑦~⑪)을 반복
                        하다 f가 0이 되면 출력을 위해 ⑬번으로 이동한다.
                        break;   while문(⑦~⑪)을 빠져나가 ⑬번으로 이동한다.
            ⑬      for (n = 1; n <= f; n++)
            {
                ⑭      k++;
                ⑮      i += c;
                ⑯      a[i][j] = k;
            }
            ⑰      c *= -1;
        }
        ⑱      for (int x = 0; x <= 4; x++)
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");
        }
    }
}

```

047 배열 10 - 대각선으로 채우기



문제 5행 5열의 배열에 다음과 같이 기억시키는 순서도를 작성하시오.

1	2	4	7	11
3	5	8	12	16
6	9	13	17	20
10	14	18	21	23
15	19	22	24	25

JAVA 코드

```

public class Section047 {
    public static void main(String[] args) {
        ❶ int i, j, k;
        ❶ int a[][] = new int[5][5];
        ❶ int L = 0;
        ❷ for (i = 0; i <= 8; i++)  배열의 위치는 0부터 시작한다. 바깥쪽 반복문은 회전수를 지정하는데, i가 0~8까지 변하는 동안 ❶~❷번을 반복 수행한다.

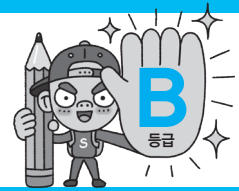
        {
            ❸ for (j = 0; j <= 4; j++)  배열의 위치가 0부터 시작하기 때문에 행 첨자 j는 0~4까지 변하는 동안 ❶~❷번을 반복 수행한다.

            {
                ❹ k = i - j;
                ❺ if (k < 0) continue;  열의 위치가 0~4까지 변하기 때문에 열이 0보다 작아지면 다음 위치 계산을 위해 인쪽 for문의 시작 위치인 ❸번으로 가고, 아니면 ❹번으로 간다.
                ❻ if (k > 4) continue;  열의 위치가 0~4까지 변하기 때문에 열이 4보다 커지면 다음 위치 계산을 위해 인쪽 for문의 시작 위치인 ❸번으로 가고, 아니면 ❷번으로 간다.

                ❼ L++;
                ❽ a[j][k] = L;
            }
        }
        ❾ for (int x = 0; x <= 4; x++)
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");
        }
    }
}

```

048 배열 11 - 마방진 (Magic Square)



문제

5행 5열의 배열 A에 다음과 같이 가로와 세로의 합이 모두 같은, 일명 마방진을 완성하는 순서도를 작성하십시오.

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

JAVA 코드

```

public class Section048 {
    public static void main(String[] args) {
        ① int i, j, k, nmg;
        ① int a[][] = new int[5][5];
        ② i = 0; 배열의 위치가 0부터 시작하기 때문에 행의 시작 위치인 i를 0으로 초기화한다.
        ③ j = 5 / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다. 배열의 위치가 0부터
            시작하기 때문에 0에서 4까지 증가하는 행의 중간 위치는 5를 2로만 나누면 된다.
        ④ for (k = 1; k <= 25; k++)
            {
        ⑤     a[i][j] = k;
        ⑥     nmg = k - k / 5 * 5;
        ⑦     if (nmg == 0)
            {
        ⑧         i++;
            continue; 제어가 for문의 시작인 ④번으로 이동한다.
            }
        ⑨     i--;
        ⑩     j++;
        ⑪     if (i < 0) 행의 위치가 0~4까지 변하기 때문에 행이 0보다 작아져 배열의 크기를 벗어나는지 검사한다.
        ⑫     i = 4; 행이 0보다 작아진 경우 행의 위치를 마지막인 4로 변경한다.
        ⑬     if (j > 4) 열의 위치가 0~4까지 변하기 때문에 열이 4보다 커져 배열의 크기를 벗어나는지 검사한다.
        ⑭     j = 0; 열이 4보다 커진 경우 열의 위치를 처음인 0으로 변경한다.
            }
        ⑮ for (int x = 0; x <= 4; x++)
            {
                for (int y = 0; y <= 4; y++)
                    System.out.printf("%3d", a[x][y]);
                System.out.printf("\n");
            }
        }
    }
}

```

049 배열 12 - 행렬 변환



기사 15.4, 12.4

문제

2차원 배열의 크기를 입력 받아 그림과 같이 숫자를 채운 후 행과 열을 바꿔 출력하는 순서도를 작성하시오. 단, 여기에서는 행으로 5를, 열로 3을 입력받았다고 가정한다.

배열 A

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

배열 B

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

JAVA 코드

```
import java.util.Scanner;

public class Section049 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int row, col, i, j, L, m;
        ② row = scanf.nextInt();
        ② col = scanf.nextInt();
        ③ int a[][] = new int[row][col];
        ④ int b[][] = new int[col][row];
        ④ int k = 0;
        ⑤ for (i = 0; i <= row - 1; i++)
            ⑥ {
                for (j = 0; j <= col - 1; j++)
                    ⑦ {
                        ⑦ k++;
                        ⑧ a[i][j] = k;
                    }
            }
        ⑨ L = 0;
        ⑩ m = -1;
    }
}
```

배열의 위치가 0부터 시작하기 때문에 행 첨자 i 는 0부터 ($row-1$)까지 변하는 동안 ⑥~⑧번을 반복 수행한다.

배열의 위치가 0부터 시작하기 때문에 열 첨자 j 는 0부터 ($col-1$)까지 변하는 동안 ⑦~⑧번을 반복 수행한다.

배열 B의 행이 0부터 시작하기 때문에 행 첨자 L 을 0으로 초기화한다.

배열 B의 열 첨자 m 은 ' $m+1$ (⑩)'을 수행한 후 0이 되도록 -1로 초기화한다.

```

11  for (i = 0; i <= row - 1; i++)  배열의 위치가 0부터 시작하기 때문에 행 첨
                                   자 i는 0부터 (row-1)까지 변하는 동안 12~
                                   17번을 반복 수행한다.
12  {
    12  for (j = 0; j <= col - 1; j++)  배열의 위치가 0부터 시작하기 때문에
                                       열 첨자 j는 0부터 (col-1)까지 변하
                                       는 동안 13~17번을 반복 수행한다.
        {
            13  m += 1;
            14  b[L][m] = a[i][j];
            15  if (m >= row - 1)  배열 B의 열을 나타내는 m이 배열 a의 행을 나타내는
                                   (row-1)보다 크거나 같으면 행을 증가시키기 위해 16
                                   번으로 가고, 아니면 17번으로 간다.
                {
                    16  L++;
                    17  m = -1;  배열 B의 열 첨자 m은 'm+=1(13)'을 수행한 후 0이 되도록 -1로 초
                                   기화한 후 제어가 12번으로 이동한다.
                }
        }
    }
18  for (int x = 0; x < col; x++)
    {
        for (int y = 0; y < row; y++)
            System.out.printf("%3d", b[x][y]);
        System.out.printf("\n");
    }
    scanf.close();
}

```



문제 8 다음은 5행 5열의 배열에 다음과 같이 숫자를 차례로 기억시키고 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

public class Test08 {
    public static void main(String[] args) {
        ①      int k, i, j, n;
               int a[][] = new int[5][5];
        ②      k = 0;
        ②      i = j = -1;   배열의 위치가 0부터 시작하기 때문에 i와 j는 ⑤번과 ⑩번을 수행한 후 0
                           이 되도록 모두 -1로 초기화한다.
        ③      n = -1;   첫 행에서 열의 증가값은 1이지만 ⑥번을 실행한 후 1이 되도록 -1로 초기화한다.
        ④      while(true)   while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기
                           전까지 ⑤~⑭번 문장을 무한 반복한다.
        {
        ⑤          i++;   i는 행을 지정한다. 행을 지정하는 i를 0에서 4까지 증가시키기 위해 1을 더한다.
        ⑥          n *= -1;   열 위치의 증가값을 1 또는 -1로 변경하기 위해 n에 -1을 곱한다.
        ⑦          if (i > 4)   행의 위치가 0~4까지 변하기 때문에 i가 4보다 커져 모든 수행이 끝
                           났는지 확인한다.
        ⑧              break;   while 반복문을 탈출한다. 제어가 ⑭번으로 이동한다.
        ⑨          while(true)   while 반복문의 시작점이다. 조건이 참(true)이므로 break를
                           만나기 전까지 ⑩~⑰번 문장을 무한 반복한다.
        {
        ⑩              j += n;   j는 열을 지정한다. 열을 지정하는 j를 0에서 4 또는 4에서 0까지
                           증가시키기 위해 j에 n을 누적한다.
        ⑪              k++;   k는 1에서 25까지 증가시키기 위해 1을 더한다.
        ⑫              a[i][j] = k;   a 배열에 i행 j열(a[i][j])에 k를 저장한다.
        ⑬              if (i % 2 == 0)   배열의 위치가 0부터 시작하기 때문에 i가 짝수이면
                           ⑩번으로 가고, 아니면 ⑰번으로 간다.
        {
        ⑭                  if (j < 4)   이곳으로 온 경우는 짝수 행으로 열의 위치가 0~4까
                           지 변하기 때문에 열이 4보다 작으면 ⑮번으로 가고,
                           아니면 ⑰번으로 간다.
        ⑮                      continue;   제어가 while문의 시작점으로 이동하여 ⑩번
                           부터 다시 시작한다.
        ⑯                  else
        {
        ⑰                      j++;   열을 나타내는 j가 4가 되었을 경우이다. 다음 행의 열 j
                           는 4~0까지 변해야 하므로 5를 만들기 위해 j에 1을 더
                           한다.
        ⑱                      break;   while 반복문을 탈출한다. 제어가 ⑭번으로 이동한다.
        }
        }
    }
}

```



```
19         else
20         {
21             if (j > 0) 이곳으로 온 경우는 홀수 행으로 열의 위치가 4~0까
                지 변화하기 때문에 열이 0보다 크면 21번으로 가고,
                아니면 23번으로 간다.
22             continue; 제어가 while문의 시작점으로 이동하여 19번
                부터 다시 시작한다.
23         else
24         {
25             j = -1; 열을 나타내는 j가 0이 되었을 경우이다. 다음 행
                의 열 j는 0~4까지 변해야 하므로 j를 -1로 초기화
                한다.
26             break; while 반복문을 탈출한다. 제어가 23번으로 이동
                한다.
                }
            }
        }
    }
    for (int x = 0; x <= 4; x++) a 배열의 내용을 출력하고 끝낸다.
    {
        for (int y = 0; y <= 4; y++)
            System.out.printf("%3d", a[x][y]);
        System.out.printf("\n");
    }
}
```



문제 9 다음은 5행 5열의 배열에 다음과 같이 숫자를 차례로 기억시키고 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test09 {
    public static void main(String[] args) {
        ① int c, n, s, j, r1, r2;
        int a[][] = new int[5][5];
        ② c = 2; 배열의 위치가 0부터 시작하기 때문에 5행 5열의 중간 행은 2이므로, 행 위치가 저장될 변수 c를 2로 초기화한다.
        ② n = s = -1; 2씩 증가되는 숫자가 저장될 변수 n은 ⑨번을 수행한 후 10이 되도록 -1로 초기화한다.
        배열의 위치가 0부터 시작하기 때문에 배열의 행 위치를 지정할 변수 s는 ⑥번을 수행한 후 0이 되도록 -1로 초기화한다.
        ③ r1 = c; 첫 번째 행에서 열의 시작 위치는 2열이므로, r1을 2(C)로 초기화한다.
        ④ r2 = c;
        ⑤ do ~while 반복문의 시작점으로 ⑥~⑬번 문장을 반복한다.
        {
            ⑥ s++; 배열의 행은 1부터 5까지 변화하므로 행을 지정하는 변수 s를 1씩 증가시킨다.
            ⑦ j = r1; 배열의 열을 지정하는 변수 j에 열의 시작 위치를 지정하는 변수 r1의 값을 저장한다.
            ⑧ do ~while 반복문의 시작점으로 ⑨~⑪번 문장을 반복한다.
            {
                ⑨ n += 2; n을 1부터 25까지 2씩 증가시키기 위해 2를 더한다.
                ⑩ a[s][j] = n; a 배열의 s행 j열(a[s][j])에 n을 저장한다.
                ⑪ j++; 열의 위치를 1씩 증가시킨다.
                ⑫ } while (j <= r2); 열의 위치는 열의 시작 위치인 r1부터 열의 끝 위치인 r2까지 변화하므로 j가 r2보다 작거나 같으면 현재 행을 다 채우지 않았으므로 ⑨번으로 가고, 아니면 다음 행을 채우기 위해 ⑬번으로 간다.
                ⑬ if (s >= c) 현재 작업한 행(s)이 2(c)보다 크거나 같으면 열의 시작 위치 r1을 1씩 증가시키고, 열의 끝 위치 r2는 -1씩 증가시키기 위해 ⑭번으로 가고, 아니면 반대의 작업을 수행하기 위해 ⑰번으로 간다.
                {
                    ⑭ r1++; 열의 시작 위치 r1을 1씩 증가시킨다.
                    ⑮ r2--; 열의 끝 위치 r2를 -1씩 증가시킨다.
                }
                ⑯ else
                {
                    ⑰ r1--; 열의 시작 위치 r1을 -1씩 증가시킨다.
                    ⑱ r2++; 열의 끝 위치 r2를 1씩 증가시킨다.
                }
            } while (s < 4); 배열의 위치가 0~4까지 변화하므로 s가 4보다 작으면 아직 채워야 할 행이 남아 있으므로 ⑥번으로 가고, 아니면 다 채운 것이므로 끝낸다.
        }
    }
}
```



문제 10 다음은 구현된 7행 7열의 배열에 다음과 같이 숫자를 차례로 기억시키고 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test10 {
    public static void main(String[] args) {
        ①      int j, s, f, q, k, n, ct;
               int a[][] = new int[7][7];
        ②      n = 7;
        ②      k = 0;
        ③      j = s = f = q = -1;  배열의 위치가 0부터 시작하기 때문에 행과 열의 위치를 지
                               정해 주는 변수들을 모두 -1로 초기화한다.
        ④      ct = n / 2;  정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함
                               수를 사용하지 않아도 된다. 배열의 위치가 0부터 시작하기 때문에 0에서 6
                               까지 증가하는 배열의 중간 열 번호는 2로만 나누면 된다.
        ⑤      for (j = 0; j <= n - 1; j++)  배열의 위치가 0부터 시작하기 때문에 열
                               첨자 j는 0부터 (n-1)까지 변하는 동안 ⑥~
                               ⑩번을 반복 수행한다.
        {
        ⑥      if (j < ct)  j가 ct(가운데 열)보다 작으면 ⑦번으로 가고, 아니면 ⑧번으로
                               간다.
        {
        ⑦      s = j;  1열에서 중간 열(ct) 전까지 사용할 행의 시작 위치다. 행의 시작 위치
                               는 열 번호 j와 같다.
        ⑧      f = n - j;  행의 끝 위치는 열이 증가할 때마다 전체 행수에서 증가한 열
                               의 수만큼 줄어든 값에 1을 더한 값만큼 변화해야 하지만 배
                               열의 위치가 0부터 시작하기 때문에 행의 끝 위치는 n-j가
                               된다.
        }
        else
        {
        ⑨      s = n - (j + 1);  중간 열 이후의 행 시작 위치는 열이 증가할 때마
                               다 전체 행수에서 증가한 열의 수만큼 줄어든 값에
                               1을 더한 값만큼 변화해야 하지만 배열의 위치가 0
                               부터 시작하기 때문에 전체 행수에서 증가한 열의
                               수에 1을 더한 수만큼 줄어들어야 한다.
        ⑩      f = j + 1;  중간 열 이후의 행 끝 위치는 열 번호 j와 같아야 하지만 배열
                               의 위치가 0부터 시작하기 때문에 j에 1을 더한 수가 되어야
                               한다.
        }
    }
}
```



예 · 상 · 문 · 제 · 은 · 행

```
11      for (q = s; q <= f - 1; q++)
12          {
13              k++;   k를 1에서 31까지 증가시키기 위해 1을 더한다.
              a[q][j] = k;   a 배열의 a행 j열(a[q][j])에 k를 저장한다.
          }
      }
  }
```

안쪽 반복문은 행을 지정한다. 행은 행의 시작 위치인 **s**에서 행의 끝 위치인 **f**여야 하지만 배열의 위치가 0부터 시작하기 때문에 **f-1**까지 반복하여 수행해야 한다.



문제 11 다음은 5행 5열의 배열에 배열의 가로 세로의 합이 모두 같은, 일명 마방진을 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

public class Test11 {
    public static void main(String[] args) {
        ① int i, j, k, nmg;
        int a[][] = new int[5][5];
        ② i = k = 0; 배열의 위치가 0부터 시작하기 때문에 행의 시작 위치를 지정해 주는 i를 0
            으로 초기화한다.
        ③ j = 5 / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수
            를 사용하지 않아도 된다. 배열의 위치가 0부터 시작하기 때문에 0에서 4까
            지 증가하는 배열의 중간 열 번호는 2로만 나누면 된다.
        do do~while 반복문의 시작점으로 ④~⑩번 문장을 반복한다.
        {
            ④ k++; k를 1에서 25까지 증가시키기 위해 1을 더한다.
            ⑤ a[i][j] = k; a 배열의 i행 j열(a[i][j])에 k를 저장한다.
            ⑥ nmg = k - k / 5 * 5; 저장된 숫자가 5의 배수인지를 확인하기 위해 나머
                지를 계산한다.
            ⑦ if (nmg == 0) nmg가 0이면 5의 배수이므로 ⑧번으로 가고, 아니면 ⑨번으로
                간다.
            ⑧ i++; 다음 행을 지정하기 위해 행을 1씩 증가시킨다.
            else
            {
                ⑨ i--; 다음 행을 지정하기 위해 행을 1씩 감소시킨다.
                ⑩ j++; 다음 열을 지정하기 위해 열을 1씩 증가시킨다.
                ⑪ if (i < 0) 행의 위치가 0~4까지 변하기 때문에 행이 0보다 작아져 배열
                    의 크기를 벗어나면 ⑫번으로 가고, 아니면 ⑬번으로 간다.
                ⑫ i = 4; 행이 0보다 작아진 경우 행의 위치를 마지막인 4로 변경한다.
                ⑬ if (j > 4) 열의 위치가 0~4까지 변화하기 때문에 열이 4보다 커져 배열
                    의 크기를 벗어나면 ⑭번으로 가고, 아니면 ⑮번으로 간다.
                ⑭ j = 0; 열이 4보다 커진 경우 열의 위치를 처음인 0으로 변경한다.
            }
        } while (k < 25); k가 25보다 작으면 다음 숫자를 채우기 위해 ④번으로 가고, 아
            니면 ⑩번으로 간다.
        ⑯ for (int x = 0; x <= 4; x++) a 배열을 출력하고 끝낸다.
        {
            for (int y = 0; y <= 4; y++)
                System.out.printf("%3d", a[x][y]);
            System.out.printf("\n");
        }
    }
}

```



문제 12 다음은 5행 5열의 배열에 직각 삼각형 형태로 1000까지의 자연수 중 2부터 시작하여 순차적으로 소수 15개까지 저장하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test12 {
    public static void main(String[] args) {
        ① int cnt, p, q, mok, nam = 0, k; JAVA에서는 변수의 값이 정의되지
            않고 사용되거나 사용될 가능성이 있는 경우 오류가 발생한다. ⑤번에
            서 cnt가 1보다 작을 경우 ⑦번을 수행하지 않고 ⑨번으로 진행되었을
            때 정의되지 않은 알 수 없는 값이 nam에 저장되어 있어 실행 시 오류
            가 발생하므로, C언어와 다르게 nam을 0으로 초기화하였다.

            int tmp[] = new int[15];
            int a[][] = new int[5][5];
        ② tmp[0] = 2; 배열의 위치가 0부터 시작하기 때문에 tmp의 첫 번째와 두 번째 요소는
            tmp[0]과 tmp[i]이다.

        ② tmp[1] = 3;
        ③ cnt = 1; 배열의 첨자로 사용하는 cnt는 두 번째 소수를 구하고 시작하는 것이므로 2로
            해야 하지만 배열의 위치가 0부터 시작하기 때문에 1로 초기화한다.

        ④ for (p = 5; p <= 1000; p += 2) 2를 제외한 짝수는 소수가 아니므로 홀
            수만을 대상으로 하기 위해 반복 변수 p가 5에서 1000이 될 때까지 2씩
            증가하면서 ⑥~⑧번을 반복 수행한다.

        {
        ⑤ for (q = 1; q <= cnt; q++) 안쪽 반복문은 대상 숫자가 소수인지를
            판별하는 것으로, 반복 변수 q를 2부터 시작해야 하지만 배열의 위치가
            0부터 시작하기 때문에 1부터 시작해서 cnt가 될 때까지 1씩 증가하면
            서 ⑥~⑧번을 반복 수행한다. q를 1부터 시작하는 이유는 tmp 배열의
            첫 번째에 2가 저장되어 있기 때문이다. 소수인지 판별할 수 p가 5, 7,
            9, 11...로 증가하는 홀수이기 때문에 2로 나눠서 나머지가 0인지 확인
            할 필요는 없다.

            {
        ⑥ mok = p / tmp[q]; 몫을 구한다.
        ⑦ nam = p - mok*tmp[q]; 나머지를 구한다.
        ⑧ if (nam == 0) 나머지가 0이면 소수가 아니므로 다음 수를 판별하기 위
            해 다음 문장으로 가고, 아니면 ⑥번으로 간다.
                break; 안쪽 반복문(for)을 탈출한다. 제어가 ⑨번으로 간다.
            }
        ⑨ if (nam == 0) ⑧번 조건을 만족한 경우 break에 의해 안쪽 반복문을 빠져나
            와 바깥쪽 반복문의 시작 위치로 이동해야 하지만 안쪽 반복문
            을 빠져나오면 처리문(⑩~⑫)이 있기 때문에 이 처리문을 수행
            하지 않고 바로 바깥쪽 반복문의 시작 위치로 이동하기 위해 한
            번 더 ⑥번의 조건을 비교한다.
```



```

    continue;   제어가 바깥쪽 반복문(for)의 시작 위치인 ⑩번으로 이동한다.
10 cnt++;   cnt를 1씩 증가시킨다.
11 tmp[cnt] = p;   tmp 배열에 소수 p를 저장한다.
12 if (cnt == 14)   배열의 위치가 0부터 시작하기 때문에 cnt는 0~14까지 15개
                  소수를 저장하기 위한 위치로 사용된다. cnt가 14이면 소수
                  15개를 모두 찾은 것이므로 ⑬번으로 가고, 아니면 바깥쪽 반
                  복문(for)의 시작 위치인 ⑩번으로 이동한다.
                  break;   바깥쪽 반복문(for)을 탈출한다. 제어가 ⑬번으로 간다.
}
13 cnt = 0;   배열의 위치가 0부터 시작하기 때문에 첫 번째 소수의 위치를 가리키는 첨자로 0
              을 사용한다.
14 for (p = 0; p <= 4; p++)   배열의 위치가 0부터 시작하기 때문에 행 첨자 p는
                              0~4까지 변하는 동안 ⑮~⑰번을 반복 수행한다.
{
15     k = 4 - p;   행 첨자 p가 0부터 시작하기 때문에 열의 시작 위치는 4-p로 지정한
                  다. 즉 0행에서는 4열부터 시작한다.
16     for (q = k; q <= 4; q++)   배열의 위치가 0부터 시작하기 때문에 열
                                  첨자 q는 k에서 시작해서 4까지 변하는 동
                                  안 ⑱~㉑번을 반복 수행한다.
    {
17         a[p][q] = tmp[cnt];   a 배열에 소수를 저장한다.
18         cnt++;   cnt를 1씩 증가시킨다.
    }
}
19 for (int x = 0; x <= 4; x++)   배열 a를 출력하고 끝낸다.
{
    for (int y = 0; y <= 4; y++)
        System.out.printf("%3d", a[x][y]);
    System.out.printf("\n");
}
}
```



문제 13 다음은 5행 5열의 배열에 배열 p의 값을 저장한 후 두 번 90도 회전하여 결과를 다시 배열 p에 이동하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
public class Test13 {
    public static void main(String[] args) {
        ①      int x, y, h, L;
                int p[][] = new int[5][5];
                int q[][] = new int[5][5];
        ②      h = L = 0;
        ③      for (x = 0; x <= 4; x++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 x는 0~4까지 변하는 동안 ④
                                   번을 반복 수행한다.
        {
            ④      for (y = 0; y <= 4; y++) 배열의 위치가 0부터 시작하기 때문에 열 첨자 y는 0~4까지 변하는
                                   동안 ⑤~⑥번을 반복 수행한다.
            {
                ⑤      h += 2; h를 2부터 50까지 2씩 증가시키기 위해 2를 더한다.
                ⑥      p[x][y] = h; p 배열에 값을 저장한다.
            }
        }
        ⑦      for (x = 2; x >= 1; x--) 한 번은 p 배열을 q 배열로 이동하고, 또 한 번은 q 배열을 p 배열로 이동하는
                                   작업을 수행하기 위해 반복 변수 x를 2에서 1이 될 때까지 -1씩 증가하면서 ⑧
                                   번을 반복 수행한다.
        {
            ⑧      for (y = 0; y <= 4; y++) 배열의 위치가 0부터 시작하기 때문에 행 첨자 y는 0~4까지 변하는
                                   동안 ⑨번을 반복 수행한다.
            {
                ⑨      for (h = 0; h <= 4; h++) 배열의 위치가 0부터 시작하기 때문에 열 첨자 h는 0~4까지
                                   변하는 동안 ⑩~⑪번을 반복 수행한다.
                {
                    ⑩      L = 4 - h; 열 첨자 h가 0부터 시작하기 때문에 이동할 배열의 행은 4-h로 지정한다. 즉 h가
                                   0~4까지 변하는 동안 L은 4~0까지 변한다.
                    ⑪      if (x == 2) p 배열을 q 배열로 이동할 지 q 배열을 p 배열로 이동할 지를 판단한다. x가 2
                                   일때는 p 배열의 값을 q 배열로 치환하기 위해 ⑫번으로 가고, 아니면 ⑬번으로
                                   간다.
                    ⑫      q[y][h] = p[L][y]; p 배열에 q 배열의 값을 치환한다.
                    ⑬      else p[y][h] = q[L][y]; q 배열에 p 배열의 값을 치환한다.
                }
            }
        }
    }
}
```



문제 14 다음은 9행 9열의 배열에 숫자를 차례로 기억시키고 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```

public class Test14 {
    public static void main(String[] args) {
        ① int q, e, c, t, j;
           int a[][] = new int[9][9];
        ② q = e = c = 0;
        ③ t = j = 0;
        ④ c = 9 / 2; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수
              를 사용하지 않아도 된다. 배열의 위치가 0부터 시작하기 때문에 0에서 4까
              지 변하는 배열의 중간 열 번호는 2로만 나누면 된다.
        ⑤ for (t = 0; t <= 8; t++) 배열의 위치가 0부터 시작하기 때문에 열 첨자 t는
              0~8까지 변하는 동안 ⑥~⑪번을 반복 수행한다.
        {
            ⑥ if (t < c) t가 c(가운데 열)보다 작으면 행의 끝 위치가 1씩 감소해야 하므로 ⑦
              번으로 가고, 아니면 행의 끝 위치가 현재 열의 위치와 같아야 하므로
              ⑧번으로 간다.
            ⑦ e = 8 - t; 열 첨자 t가 0부터 시작하기 때문에 행의 끝 위치인 e는 8-t
              로 지정한다. 즉 t가 0~8까지 변하는 동안 e는 8~0까지 변
              한다.
            else
            ⑧ e = t; 행의 끝 위치를 계산한다. t(현재 열)이 c(가운데 열)보다 작지 않은
              경우로 t값을 행의 끝 위치로 사용한다.
            ⑨ for (j = 0; j <= e; j++) 배열의 위치가 0부터 시작하기 때문에 행
              첨자 j는 0에서 시작해서 e까지 변하는 동
              안 ⑩~⑪번을 반복 수행한다.
            {
            ⑩ q++; q를 1에서 65까지 증가시키기 위해 1을 더한다.
            ⑪ a[j][t] = q; a 배열의 j행 t열(a[j][t])에 q를 저장한다.
            }
        }
    }
}

```

050 화폐의 종류별 매수 계산



기사 19.6, 17.4, 15.10, 12.10, 06.11

문제

다음 입력 형식과 같이 입력 받은 후 화폐 단위별로 화폐의 매수를 계산하여 출력 형식에 맞게 출력하는 순서도를 작성하시오.

〈입력 형식〉

이름	출장비 지급액
----	---------

〈출력 형식〉

출장비 지급표

성명	출장비지급액	오만	일만	오천	일천	오백	일백	오십	일십	오	일
엄철식	XXXX	X	X	X	X	X	X	X	X	X	X
양동수	XXXX	X	X	X	X	X	X	X	X	X	X
이동훈	XXXX	X	X	X	X	X	X	X	X	X	X
전체 화폐 매수		X	X	X	X	X	X	X	X	X	X

〈처리 조건〉

1. 이름으로 "QUIT"가 입력되면 종료한다.

JAVA 코드

```

import java.io.FileReader;      FileReader를 사용하기 위한 클래스이다.
import java.io.IOException;     파일 입출력 과정에서 발생하는 예외를 처리하기 위한 클래스이다.
import java.util.Scanner;

public class Section050 {
    static final int LENGTH = 10;  프로그램 안에서 LENGTH는 항상 고정된 값 10을 갖도록
                                   정의한다.

    static class member { member 클래스를 정의한다. JAVA는 구조체 대신 클래스를 활용하
                           여 구조체와 동일하게 사용하는 것이 가능하다.

        String n;
        int p;
        int pm[] = new int[10];
    }

    public static void main(String[] args) throws IOException {
        입출력 과정에서 발생하는 예외 처리를 IOException 클래스로 넘긴다. JAVA에서는 파일 입출력 시 예외 처
        리를 하지 않으면 오류가 발생한다.
        Scanner inf = new Scanner(new FileReader("data1.txt"));
        Scanner 클래스의 객체 변수 inf를 생성하면서 FileReader를 사용해 'data1.txt' 파일로부터
        읽어오도록 만든다.
        ❶ member data = new member(); member 객체 변수 data를 선언한다. 접근은 C언
            어와 동일하다.

        ❶ int tm[] = new int[10];
        ❶ int k, t, m, sw;
        ❷ System.out.printf("\t\t\t\t\t출장비 지급표\n"); <출력 형식>에 맞게 제목을
            출력한다.

        ❷ System.out.printf("성명 출장비지급액 오만 일만 오천 일천 오백 일백 오십 일십 오
            일\n"); <출력 형식>에 맞게 필드명을 출력한다.

        ❸ for (k = 0; k <= LENGTH - 1; k++) 배열의 위치를 0부터 시작하기 때문
            에 첨자의 초기값과 최종값이 순서도와 다르다.
            LENGTH는 10이므로 첨자 k는 0~9까지 변하는
            동안 ❹번을 반복 수행한다.

        ❹ tm[k] = data.pm[k] = 0; 'tm[k]=0', 'data.pm[k]=0'과 동일하다.
        ❺ data.n = inf.next(); 파일에서 단어를 읽어 객체 변수 data에 있는 문자열 변수 n에
            저장한다.
        ❺ data.p = inf.nextInt(); 파일에서 숫자를 읽어 객체 변수 data에 있는 정수 변수
            p에 저장한다.
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
            지 ❹~❷번 문장을 무한 반복한다.
        {
            ❻ t = data.p;
            ❼ m = 50000;
            ❸ sw = 1;
            ❹ for (k = 0; k <= LENGTH - 1; k++) LENGTH는 10이므로 첨자 k는
                0~9까지 변하는 동안 ❶~❷번
                을 반복하여 수행한다.

            ❶ {
                ❶ data.pm[k] = t / m; 정수형 변수는 소수점 이하는 버리고 정수만 저
                    장하므로 정수를 구하는 함수를 사용하지 않아도
                    된다.
            }
        }
    }
}

```

```

11         t -= data.pm[k] * m;
12         if (sw == 1)  SW가 1이면 13~14번을 실행하고, 아니면 15~16번을 실행한다.
            {
13             m /= 5;
14             sw = 0;
            }
            else
            {
15                 m /= 2;
16                 sw = 1;
            }
        }
17     System.out.printf("%s %d", data.n, data.p);  data.n의 값
                                                    을 문자열로 출력하고, data.p
                                                    의 값을 정수로 출력한다.
18     for (k = 0; k <= LENGTH - 1; k++)  LENGTH는 100이므로 첨자 k는
                                                    0~9까지 변하는 동안 19~20번
                                                    을 반복 수행한다.
        {
19             System.out.printf("%5d", data.pm[k]);  data.pm[k]의
                                                    값을 5칸에 정수로 출력한다.
20             tm[k] += data.pm[k];
        }
        System.out.printf("\n");  정렬을 위해 한 줄을 띄운다.
21     data.n = inf.next();
22     if(data.n.equals("quit")) break;  JAVA에서는 EOF(End of
                                        File)일 때 파일을 더 읽을 경우 오류가 발생한다.
                                        때문에 'quit'를 읽은 지점 바로 뒤에서 조건문을
                                        통해 while문을 빠져나간다.
                                        문자열 클래스의 메소드 equals를 사용해 data.
                                        n이 "quit"와 동일한지 비교하고, 동일하면 23번으
                                        로 이동한다.
23         data.p = inf.nextInt();
    }
24     System.out.printf("전체 화폐 매수 ");
25     for (k = 0; k <= LENGTH - 1; k++)  LENGTH는 100이므로 첨자 k는 0~9까
                                                    지 변하는 동안 26번을 반복 수행한다.
        System.out.printf("%5d", tm[k]);  tm[k]의 값을 5칸에 정수로 출력
                                                    한다.
    inf.close();  객체 변수 inf를 닫는다. 파일 읽는 기능의 객체를 닫을 경우 메모리가 확보되
                고, 쓰는 기능의 객체를 닫을 경우 확실하게 저장하기 위함이다.
}

```

051 부서별 합계



문제

다음 입력 형식과 같이 입력 받은 후 출력 형식에 맞게 출력하는 순서도를 작성하시오.

〈입력 형식〉

부서	성명	본봉	수당
----	----	----	----

〈출력 형식〉

사원 급여표

부서	성명	본봉	수당	합계
영업	×××	××	××	××
영업	×××	××	××	××
부서 합계 :	×××			
총무	×××	××	××	××
총무	×××	××	××	××
부서 합계 :	×××			
전체 합계 :	×××			

〈처리 조건〉

1. 입력 자료는 부서별로 정렬되어 있다.
2. 부서가 변경되면 부서의 합계를 출력한다.
3. 자료의 끝(Eof)*인 경우 전체 합계를 출력한다.

JAVA 코드

```

import java.io.FileReader;  FileReader를 사용하기 위한 클래스이다.
import java.io.FileWriter; FileWriter를 사용하기 위한 클래스이다.
import java.io.IOException; 파일 입출력 과정에서 발생하는 예외를 처리하기 위한 클래스이다.
import java.util.Scanner;

public class Section051 {
    static class member { member 클래스를 정의한다. JAVA는 구조체 대신 클래스를 활용하
                           여 구조체와 동일하게 사용하는 것이 가능하다.

        String bu, irum;
        int bon, su;
    }
    public static void main(String[] args) throws IOException {
        Scanner inf = new Scanner(new FileReader("data2.txt"));
        FileWriter ouf = new FileWriter("result2.txt");
        FileWriter 클래스의 객체 변수 ouf를 선언하고, 'result2.txt' 파일을 연다. 파일이 없을 경우
        새로 생성된다.
        ❶ member data = new member(); member 객체 변수 data를 선언한다.
        ❶ int gtot, butot, keb;
        ❶ String bubi;
        ❷ gtot = 0;
        ❸ ouf.write(String.format("\t\t사원 급여표\r\n")); 제목을 출력한다.
            • ouf.write(): () 괄호 안의 내용을 FileWriter 변
              수인 ouf에 연결되어 있는 파일에 출력한다.
            • String.format(): printf와 동일하게 서식을 지정
              해주는 메소드이다.
            • "\r": JAVA에서 파일 출력 시 줄을 나눌 때 개행문자
              (Carriage Return)를 사용하지 않으면, 운영체제에 따
              라 줄이 겹쳐 보일 수 있다.
        ❸ ouf.write(String.format("부서\t성명\t본봉\t수당\t합계\r\n"));
        ❹ data.bu = inf.next(); 파일에서 자료를 읽어서 member클래스의 객체 변수 data에
              저장한다.
        ❹ data.irum = inf.next();
        ❹ data.bon = inf.nextInt();
        ❹ data.su = inf.nextInt();
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
              지 { } 사이의 문장을 무한 반복한다.
        {
            ❺ butot = 0;
            ❻ bubi = data.bu;
            do do-while 반복문의 시작점으로 ❷~❹번 문장을 반복한다.
            {
                ❷ keb = data.bon + data.su;
                ❸ butot += keb;
                ❹ ouf.write(String.format("%s\t%s\t%d\t%d\t%d\r\n", data.bu, data.irum, data.bon, data.su,
                    keb));
                ❶ if (!inf.hasNext()) Scanner의 hasNext 메소드는 다음 자료가 비
                    어있을 때 False를 반환하기 때문에 부정 기호(!)
                    를 넣는다. 읽어 들인 데이터가 없는 것으로 EOF를
    
```

판단하는 것이 아니기 때문에 마지막 데이터를 처리하기 위해 ⑩번
과 ⑪번의 순서를 바꾼다.

```

15         {
16             gtot += butot;
17             ouf.write(String.format("부서 합계 : %d\r\n", butot));
            ouf.write(String.format("전체 합계 : %d", gtot));
            inf.close(); 객체 변수 inf와 ouf를 닫는다. 파일 읽는 기능의 객체를 닫을 경우 메
                        모리가 확보되고, 쓰는 기능의 객체를 닫을 경우 확실하게 저장하기 위함
                        이다.
            ouf.close();
            System.exit(0); 프로그램을 종료한다. JAVA에서 exit는 System 클래스에 속해
                        있다.
        }
10     data.bu = inf.next(); 파일에서 자료를 읽어서 member클래스의 객체 변수 data에
                        저장한다.
10     data.irum = inf.next();
10     data.bon = inf.nextInt();
10     data.su = inf.nextInt();
12 } while (bubi.equals(data.bu)); 문자열 클래스의 메소드 equals를 사용해 bubi
                        가 data.bu와 동일한지 비교하고, 동일하면 ⑦번
                        으로 이동한다.
13     gtot += butot;
14     ouf.write(String.format("부서 합계 : %d\r\n", butot));
    }
}

```

052 동별, 나이별 인원 통계



문제 다음의 처리 조건에 따라 동별, 나이별 통계를 구하는 순서도를 작성하시오.

〈입력 형식〉

동 코드	성명	나이
DONG	IRUM	NAI

〈출력 형식〉

동	0~9	10~19	20~29	30~39	40~49	50~59	60 이상	합계
1								
2								
3								
4								
⋮								
합계								

〈처리 조건〉

- 동은 1~10까지 10개의 동이 있다.
- 동 코드로 0이 입력되면 종료한다.
- 〈출력 형식〉에서 색상으로 칠해진 부분은 이해를 위한 것으로 실제 출력은 색상으로 칠해진 부분을 제외하고 본문의 값들만 출력하도록 한다.

JAVA 코드

```
import java.io.FileWriter;    FileWriter를 사용하기 위한 클래스이다.
import java.io.IOException;   파일 입출력 과정에서 발생하는 예외를 처리하기 위한 클래스이다.
import java.util.Scanner;

public class Section052 {
    static class member {    member 클래스를 정의한다.
        int dong, nai;
        String irum;
    }
    public static void main(String[] args) throws IOException {
        Scanner scanf = new Scanner(System.in);
        FileWriter ouf = new FileWriter("result3.txt");
        FileWriter 클래스의 객체 변수 ouf를 선언하고, 'result3.txt' 파일을 연다. 파일이 없을 경우
        새로 생성된다.
        ❶ member data = new member();    member 객체 변수 data를 선언한다.
        ❷ int a[][] = new int[11][8];
```

```

1  int i, j, row, col;
2  for (i = 0; i <= 10; i++)  JAVA에서 객체를 선언할 때 특별한 값을 저장하지 않
                               으면 정수형인 경우 0으로 초기화되므로, 배열을 초기
                               화하는 과정인 ②~④번은 생략해도 된다.
3      for (j = 0; j <= 7; j++)
4          a[i][j] = 0;
   System.out.printf("동 이름 나이를 입력하세요. Wn모두 입력하셨으면 '0 0 0'
   을 입력하세요\n");
   화면으로부터 값을 입력받고, 모두 입력받았을 시 입력받은 변수가 3개이기 때문에 '0 0 0'을 입력받는다.
   while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
                   지 { } 사이의 문장을 무한 반복한다.
   {
5       data.dong = scanf.nextInt();  화면으로부터 자료를 읽어서 객체 변수
                                       data의 변수 dong, irum, nai에 저장
                                       한다.

6       data.irum = scanf.next();
7       data.nai = scanf.nextInt();
8       if (data.dong == 0)  data.dong이 0과 같으면 다음 문장을 실행하고, 아니면
                               ⑦번으로 이동해서 실행을 계속한다.
                               break;  while 반복문을 탈출한다. 제어가 ⑥번으로 이동한다.
9       if (data.nai >= 60)  data.nai가 60보다 크거나 같으면 ⑨번을 실행하고, 아
                               니면 ⑩번으로 이동해서 실행을 계속한다.
10          data.nai = 60;
11          row = data.dong - 1;  배열의 위치가 0부터 시작하기 때문에 row의 값이 순서
                                  도와 다르다.
12          col = data.nai / 10;  배열의 위치가 0부터 시작하기 때문에 col의 값이 순서
                                  도와 다르다. 정수형 변수는 소수점 이하는 버리고 정수만
                                  저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.
13          a[row][col] += 1;
14          a[row][7] += 1;  배열의 위치가 0부터 시작하기 때문에 열 첨자의 값이 순서도와
                              다르다.
15          a[10][col] += 1;  배열의 위치가 0부터 시작하기 때문에 행 첨자의 값이 순서도와
                              다르다.
16          a[10][7] += 1;  배열의 위치가 0부터 시작하기 때문에 행 첨자와 열 첨자의 값이 순
                              서도와 다르다.
   }
17  for (i = 0; i <= 10; i++)  11행을 모두 출력하기 위해 행 첨자 i는 0에서 10까지
                              변하는 동안 반복 수행한다.
   {
18      for (j = 0; j <= 7; j++)  각 행의 8열을 모두 출력하기 위해 열 첨자 j는 0
                                  에서 7까지 변하는 동안 반복 수행한다.
19          ouf.write(String.format("%d", a[i][j]));  ouf가 가
                                                         리키고 있는 파일에 a[i][j]를 출력한다.
20          ouf.write(String.format("\r\n"));  ouf가 가리키고 있는 파일에서
                                                  커서를 다음 줄로 이동한다.
   }
   scanf.close();
   ouf.close();  객체 변수 ouf를 닫는다. 파일 읽는 기능의 객체를 닫을 경우 메모리가 확보되
                   고, 쓰는 기능의 객체를 닫을 경우 확실하게 저장하기 위함이다.
   }
}

```

053 학급별 최대, 최소 체중



문제

다음 입력 형식과 같이 입력 받은 후 출력 형식에 맞게 출력하는 순서도를 작성하시오.

〈입력 형식〉

반	번호	체중
---	----	----

〈출력 형식〉

반	번호	체중
1반	xxx	xx
	xxx	xx
	xxx	xx
4반	xxx	xx
	xxx	xx
3반	xxx	xx
	xxx	xx
	xxx	xx
최소 체중:	xxx	
최대 체중:	xxx	

〈처리 조건〉

1. 자료의 입력은 반별로 한다.
2. 반 이름이 같으면 반 이름을 출력하지 않는다.
3. 모든 자료의 출력이 끝나면 전체의 최소 체중과 최대 체중을 출력한다.
4. 번호에 0이 입력되면 종료한다.

JAVA 코드

```
import java.util.Scanner;

public class Section053 {
    static class member { member 클래스를 정의한다.
        String ban;
        int bunho, wgt;
    }
}
```

```

public static void main(String[] args) {
    Scanner scanf = new Scanner(System.in);
    ① member data[] = new member[100]; member 클래스의 크기를 갖는 100개짜리 배열 data를 선언한다.
    for (int i = 0; i < data.length; i++) 배열 data의 각 요소마다 member 객체를 선언한다.
                                         JAVA에서는 member 크기를 갖는 여러 요소들을 포함하는 배열을 선언한
                                         뒤, 배열의 각 요소에 객체 변수를 선언해 주어야 한다.

        data[i] = new member();
    ① int flag, a = 0, b = 0, i, k; JAVA에서는 변수의 값이 정의되지 않고 사용되거나 사용될 가능성이 있는 경우 오류가 발
        생한다. ⑩번에서 flag가 0이 아닌 경우 ⑫, ⑬번을 수행하지 않고 ⑮번 또는 ⑯번으로 진행되었을 때 정의되지 않은 알 수 없는
        값이 a와 b에 저장되어 있어 오류가 발생하므로, C언어와 다르게 a와 b를 0으로 초기화하였다.
    ① String c = "0번";
        System.out.printf("반 번호 무게를 입력하세요.\n모두 입력하셨으면 '0 0 0'을 입력하세요\n");
                                         화면으로부터 값을 입력받고, 모두 입력받았을 시 입력받을 변수가 3개이기 때문에 '0 0 0'을 입력받는다.

    ② flag = 0;
    ② i = k = -1; 배열의 위치를 0부터 시작하기 때문에 배열의 첨자가 되는 i와 k의 초기값이 순서도와 다르다.
    do { do~while 반복문의 시작점으로 ③~④번 문장을 반복한다.
        ③ k++;
        ④ data[k].ban = scanf.next(); 화면으로부터 자료를 읽어서 객체 배열 data[k]에 저장한다.
        ④ data[k].bunho = scanf.nextInt();
        ④ data[k].wgt = scanf.nextInt();
        ⑤ } while (data[k].bunho != 0); data[k].bunho가 0이 아닌 동안 ③~④번을 반복하여 수행한다.
        ⑥ System.out.printf("반\t번호\t체중\n");
        ⑦ while (i < k - 1) while 반복문의 시작점으로 i가 k-1보다 작은 동안 ⑨~⑮번을 반복 수행한다.
        {
            ⑨ i++;
            ⑩ if (flag == 0) flag가 0과 같으면 ⑪~⑬번을 실행하고, 아니면 ⑮번으로 이동해서 실행을 계속한다.
            {
                ⑪ flag = 1;
                ⑫ a = data[i].wgt;
                ⑬ b = data[i].wgt;
            }
            ⑯ else if (a > data[i].wgt) a가 data[i].wgt 보다 크면 다음 문장을 실행하고, 아니면 ⑮번으로 이동해서 실행을
            계속한다.
            ⑰ a = data[i].wgt;
            ⑱ else if (b < data[i].wgt) b가 data[i].wgt 보다 작으면 다음 문장을 실행하고, 아니면 ⑮번으로 이동해서 실행을
            계속한다.
            ⑲ b = data[i].wgt;
            ⑳ if (c.equals(data[i].ban)) 문자열 클래스의 메소드 equals를 사용해 c가 data[i].ban과 동일한지 비교하고,
            동일하면 ㉑번으로 이동한다.
            {
                ㉑ System.out.printf(" \t%d\t%d\n", data[i].bunho, data[i].wgt);
                continue; 이후의 문장, 즉 ㉒~㉓번을 생략하고 반복문의 처음인 ㉑번으로 이동한다.
            }
            ㉒ c = data[i].ban;
            ㉓ System.out.printf("%s\t%d\t%d\n", data[i].ban, data[i].bunho, data[i].wgt);
        }
    ⑧ System.out.printf("최소 체중:%d\n", a);
    ⑨ System.out.printf("최대 체중:%d", b);
        scanf.close();
    }
}

```

054 사과 구입



문제

3000원을 가지고 한 개에 200원, 300원, 400원인 사과를 각각 10개 이내에서 적절하게 조합하여 살 수 있는 순서도를 작성하시오. 단, 200원, 300원, 400원짜리 사과를 각각 1개 이상 사야 되며, 전체 금액은 3000원이어야 한다.

JAVA 코드

```

public class Section054 {
    public static void main(String[] args) {
        ❶ int i, j, k, cnt, hap;
        ❷ cnt = 0;
        ❸ for (i = 1; i <= 10; i++) 반복 변수 i가 1에서 시작하여 1씩 증가하면서 10보다
                                   작거나 같은 동안 ❹~❺번 문장을 10회 반복한다.
        {
            ❹ for (j = 1; j <= 10; j++) 반복 변수 j가 1에서 시작하여 1씩 증가하면서
                                   10보다 작거나 작은 동안 ❻~❽번 문장을 10회
                                   반복한다.
            {
                ❺ for (k = 1; k <= 10; k++) 반복 변수 k가 1에서 시작하여 1씩 증가
                                   하면서 10보다 작거나 작은 동안 ❻~❽
                                   번 문장을 10회 반복한다.
                {
                    ❻ hap = i * 200 + j * 300 + k * 400;
                    ❼ if (hap == 3000) hap이 3000이면 ❸, ❹번 문장을 수행하고, 아니
                                   면 { }를 빠져나간다. 제어가 for문의 시작점인
                                   ❸번으로 이동한다.
                    {
                        ❽ cnt++;
                        ❽ System.out.printf("%d %d %d %d\n", cnt,
                                   i, j, k);
                    }
                }
            }
        }
    }
}

```

055 사과 나눠 갖기



문제

A, B, C, D의 네 사람과 사과 한 바구니가 있다. A가 4등분 하였더니 1개가 남았다. 남은 1개를 먹고 4등분 한 한몫을 가졌다. B가 남은 사과를 다시 4등분 하였더니 역시 1개가 남았다. 남은 1개를 먹고 그 한몫을 가졌다. C, D도 이와 같은 방법으로 한 몫씩 가졌다. 마지막 D가 자기 몫을 가지고 간 후 나머지를 4등분 했더니 역시 1개가 남았다. 최후에 사과는 몇 개가 있었을까? 위의 조건을 만족하는 가장 작은 사과 개수를 계산하는 순서도를 작성하시오.

JAVA 코드

```

public class Section055 {
    public static void main(String[] args) {
        ① int i, j, k, nmg;
        ② i = 0;
        A: break의 위치값이다. '라벨명' 형태로 작성되며, 이 문장은 'A'라는 라벨의 break 위치이다.
        while (true)
        {
            ③ i++;
            ④ j = i;
            ⑤ for (k = 1; k <= 5; k++) 반복 변수 k가 1에서 시작하여 1씩 증가하면서 5
                                   보다 작거나 같은 동안 { } 사이의 문장을 5회 반복한다.

            ⑥ { nmg = j - j / 4 * 4; 정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.

            ⑦ if (nmg != 1) nmg가 1이 아니면 다음 문장을 수행하고, 맞으면 ⑧번으로 이동한다.
                                   break; for문을 빠져나가 while문의 처음인 ③번으로 이동한다.

            ⑧ j--;
            ⑨ j = j * 3 / 4;
            if(k == 5)
                break A; C언어와 달리 goto문이 없어 break를 활용하였다. 라벨의 위치로 바로 이동하는 goto문과는 달리 'break (라벨)'는 라벨이 있는 수준의 밖으로 탈출하게 해준다. 즉 이 문장에서는 'A' 라벨이 'while (true)' 밖에 있으므로 while문을 벗어나 ⑩번으로 이동한다.

        }
    }
    ⑩ System.out.printf("%d", i); 결과 1021
}

```

056 반 배정



문제

1번부터 21번까지 21명의 학생을 3학급으로 편성하는 순서도를 작성하시오. 첫 학생의 반은 2명의 교사가 제시한 2개의 숫자를 더한 후 21로 나눈 나머지 번째에 있는 학생을 1반으로 지정한다. 이후에는 첫 번째 지정한 학생으로부터 4명 건너뛴 학생이 2반이 되며, 다시 4명 건너뛴 학생이 3반이 된다. 이러한 방법으로 3반 다음에는 다시 1반으로 연결된다. 단, 나머지가 0이면 21로 간주하고 이미 반이 배정된 학생은 건너뛰는 4명에서 제외시킨다.

예 2명의 교사가 5와 18을 지정했을 경우 다음과 같이 지정된다.

1반 : 2, 3, 4, 7, 11, 14, 15

2반 : 6, 8, 9, 12, 13, 18, 20

3반 : 1, 5, 10, 16, 17, 19, 21

JAVA 코드

```
import java.util.Scanner;

public class Section056 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int i, x1, x2, hap, nmg, cnt, ban;
        ❷ int a[] = new int[21];
        ❸ for (i = 0; i <= 20; i++) 배열의 위치를 0부터 시작하기 때문에 첨자의 초기값과
                                최종값이 순서도와 다르다.
        ❹     a[i] = 0;
        ❺     x1 = scanf.nextInt();
        ❻     x2 = scanf.nextInt();
        ❼     hap = x1 + x2;
        ❽     nmg = hap - hap / 21 * 21; 정수형 변수는 소수점 이하는 버리고 정수만 저장하
                                므로 정수를 구하는 함수를 사용하지 않아도 된다.
        ❾     if (nmg == 0) nmg가 0이면 다음 문장을 실행하고, 아니면 ❸번 문장으로 이동해서 실행을
                                계속한다.
        ❿         nmg = 21;
        ⓫         cnt = 0;
        ⓬         ban = 0;
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
                                지 { } 사이의 문장을 무한 반복한다.
        {
            ⓭         ban++;
        }
    }
}
```

```

12     if (ban > 3)    ban이 3보다 크면 다음 문장을 실행하고, 아니면 14번 문장으로 이동해
                    서 실행을 계속한다.
13         ban = 1;
14     a[nmg - 1] = ban;    배열의 위치를 0부터 시작하기 때문에 첨자의 값이 순서도와
                        다르다.
15     cnt++;
16     if (cnt >= 21)    cnt가 21보다 크거나 같으면 다음 문장을 실행하고, 아니면 18번 문
                        장으로 이동해서 실행을 계속한다.
                        break;    while 반복문을 탈출한다. 제어가 17번으로 이동한다.
18     for (i = 1; i <= 4; i++)    반복 변수 i가 1에서 시작하여 1씩 증가하면서 4
                        보다 작거나 같은 동안 19~22번 문장을 4회 반복
                        한다.
        {
19             nmg++;
20             if (nmg > 21)    nmg가 21보다 크면 다음 문장을 실행하고, 아니면 22번 문장으
                        로 이동해서 실행을 계속한다.
21                 nmg = 1;
22             if (a[nmg - 1] != 0)    a[nmg - 1]가 0과 같지 않으면 다음 문장을 실행
                        하고, 아니면 while문의 시작점으로 이동하여
                        11번부터 다시 시작한다. 배열의 위치를 0부터 시
                        작하기 때문에 첨자의 값이 순서도와 다르다.

23                 i--;
        }
    }
17     for (i = 0; i < 21; i++)    배열 a를 모두 출력하기 위해 첨자 i는 0~20까지 21회
                        반복 수행한다.
        System.out.printf("%d ", a[i]);
        scanf.close();
    }
}

```

057 역순으로 숫자 더하기



문제

데이터를 읽어서 역순으로 변환한 후 원래의 데이터와 더하여 출력하는 순서도를 작성하시오. 예를 들어 123456이 입력되면 654321로 변환한 후 원래의 수와 더하여 출력한다. 즉, $123456 + 654321 = 777777$ 이 출력된다. 단, 입력되는 값은 6자리 정수이고 999999가 입력되면 종료한다.

JAVA 코드

```

import java.util.Scanner;

public class Section057 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ❶ int x, i, k, mok, nmg, m, j;
        ❶ int a[] = new int[6];
        while (true)  while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까
                       지 { } 사이의 문장을 무한 반복한다.
        {
            ❷ x = scanf.nextInt();
            ❸ if (x == 999999)  x가 999999와 같으면 다음 문장을 실행하고, 아니면 ❹번 문장
                           으로 이동해서 실행을 계속한다.
                           break;
            ❹ k = x;
            ❺ for (i = 0; i <= 5; i++)  배열의 위치가 0부터 시작하기 때문에 첨자 i는
                                       0~5까지 변하는 동안 ❹~❺번을 반복 수행한다.
            {
                ❻ mok = x / 10;  정수형 변수는 소수점 이하는 버리고 정수만 저장하므로 정수
                               를 구하는 함수를 사용하지 않아도 된다.
                ❼ nmg = x - mok * 10;
                ❽ a[i] = nmg;
                ❾ x = mok;
            }
            ❿ m = 0;
            ⓫ for (i = 0; i <= 5; i++)  배열의 위치가 0부터 시작하기 때문에 첨자 i는
                                       0~5까지 변하는 동안 ⓫번을 반복 수행한다.
            {
                ⓬ if (a[i] != 0)  a[i]가 0과 같지 않으면 다음 문장을 실행하고, 아니면 ⓭번
                               문장으로 이동해서 실행을 계속한다.
                               break;  for 반복문을 탈출한다. 제어가 ⓬번으로 이동한다.
            }
            ⓭ for (j = i; j <= 4; j++)  배열의 위치가 0부터 시작하기 때문에 첨자 i의
                                       최종값은 4가 된다.
            {
                ⓮ m += a[j];
                ⓯ m *= 10;
            }
            ⓰ m += a[5];
            ⓱ x = m + k;
            ⓲ System.out.printf("%d %d %d\n", k, m, x);
                           k, m, x의 값을 정수로 출력한다.
        }
        scanf.close();
    }
}

```

058 숫자의 좌우 위치 변경



상기 15.7, 12.4, 기사 19.6

문제

A[10]의 배열에는 첫 번째 위치부터 특정 번째까지 0이 아닌 자연수가 들어 있고, 나머지는 0이 들어 있다. 처음으로 0이 나오기 전까지의 숫자들의 위치를 반대로 교환하여 출력하는 순서도를 작성하시오. 예를 들어 1234500000이면 5432100000으로 출력하고, 1234000000이면 4321000000으로 출력해야 한다.

JAVA 코드

```
public class Section058 {
    public static void main(String[] args) {
        ① int i, j, m, k;
        ① int a[] = { 1,2,3,4,5,0,0,0,0,0 }; 배열 A에는 데이터가 저장되어 있다고 가정한다.
        ② i = 0; 배열의 위치를 0부터 시작하기 때문에 배열 a의 첨자가 될 i의 초기값이 순서도와 다르다.
        ③ while (a[i] != 0) a[i]가 0이 아닌 동안 ④번 문장을 반복 수행한다.
        ④ i++;
        ⑤ j = i - 1;
        ⑥ i = 0; 배열의 위치를 0부터 시작하기 때문에 배열 a의 첨자가 될 i의 초기값이 순서도와 다르다.
        ⑦ m = (i + j) / 2; 정수형 변수는 소수점 이하는 버리고 정수로 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.
        while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 { } 사이의 문장을 무한 반복한다.
        {
            ⑧ k = a[i];
            ⑨ a[i] = a[j];
            ⑩ a[j] = k;
            ⑪ if (i == m) i가 m과 같으면 다음 문장을 실행하고, 아니면 ⑫번 문장으로 이동해서 실행을 계속한다.
                        break; while 반복문을 탈출한다. 제어가 ⑬번으로 이동한다.
            ⑫ i++;
            ⑬ j--;
        }
        ⑭ for (i = 0; i <= 9; i++) 배열 a를 모두 출력하기 위해 첨자 i는 0~9까지 변화하는 동안 ⑮번을 반복 수행한다.
        ⑮ System.out.printf("%d", a[i]); a[i]의 값을 정수로 출력한다.
    }
}
```

059 구구단



상기 19.4, 18.6

문제 1단부터 9단까지의 구구단을 출력하는 순서도를 작성하시오.

JAVA 코드

```

public class Section059 {
    public static void main(String[] args) {
        ① int b, c, i;
        ① int a[] = new int[9];
        ② System.out.printf("****구구단****\n");
        ③ b = 0;
        do do~while 반복문의 시작점으로 ④~⑩번 문장을 반복한다.
        {
            ④ b++;
            ⑤ c = 0;
            do do~while 반복문의 시작점으로 ⑥~⑦번 문장을 반복한다.
            {
                ⑥ c++;
                ⑦ a[c - 1] = b*c; 배열의 위치를 0부터 시작하기 때문에 첨자의 값이 순서도와 다르다.
            } while (c < 9); c가 9보다 작은 동안 ⑥~⑦번을 반복하여 수행한다.
            ⑧ for (i = 1; i <= 9; i++) 반복 변수 i가 1에서 시작하여 1씩 증가하면서 9
            ⑨ 보다 작거나 같은 동안 ⑩번 문장을 9회 반복한다.
            ⑩ System.out.printf("%dX%d=%d\n", b, i, a[i - 1]);
            배열의 위치를 0부터 시작하기 때문에 첨자의 값이 순서도와 다르다.
            ⑪ } while (b < 9);
        }
    }
}

```



문제 3 다음은 금액을 입력받아 화폐 단위별 매수를 구한 후 결과를 출력하는 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
import java.util.Scanner;

public class Test03 {
    public static void main(String[] args) {
        Scanner scanf = new Scanner(System.in);
        ① int i, sw, j, k;
        ② i = 50000; 처음에 계산할 화폐 단위는 50000으로 i를 초기화한다.
        ③ sw = 0; 다음에 계산할 화폐 단위는 10000으로, 50000을 5로 나누기 위해 sw를 0으로 초기화한다.
        ④ j = scanf.nextInt(); 금액을 입력받는다.
        ⑤ while (true) while 반복문의 시작점이다. 조건이 참(true)이므로 break를 만나기 전까지 ⑥~⑮번의 문장을 무한 반복한다.
        {
            ⑥ k = j / i; 정수형 변수는 정수만 저장하므로 정수를 구하는 함수를 사용하지 않아도 된다.
            ⑦ System.out.printf("%d, %d\n", i, k); 화폐 단위와 매수를 출력한다.
            ⑧ if (i <= 1) i가 1보다 작거나 같으면 매수 계산이 끝난 것이므로 프로그램을 종료하고 아니면 다음 매수 계산을 위해 ⑨번으로 간다.
                break; 반복문(while)을 빠져나가 ⑥번으로 이동한다.
            ⑨ j -= k * i; 화폐 매수를 구한 금액을 뺀 나머지 금액을 계산한다.
            ⑩ if (sw == 0) sw가 0이면 화폐 단위를 5로 나누기 위해 ⑪번으로 가고, 아니면 2로 나누기 위해 ⑫번으로 간다.
                {
                    ⑪ i /= 5; 10000, 1000, 100, 10, 1에 대한 화폐 단위를 구한다.
                    ⑫ sw = 1; 다음 번에는 2로 나누기 위해 sw를 1로 치환한다.
                }
            ⑬ else
            {
                ⑭ i /= 2; 5000, 500, 50, 5에 대한 화폐 단위를 구한다.
                ⑮ sw = 0; 다음 번에는 5로 나누기 위해 sw를 0으로 치환한다.
            }
        }
        ⑯ scanf.close();
    }
}
```



산기 19년 4월, 18년 6월

문제 4 다음은 구구단 프로그램을 JAVA 언어로 구현한 것이다.

JAVA 코드

```
#package test;
```

```
public class Test {  
    public static void main(String[ ] args) {
```

① `int i, j;` 정수형 변수 `i, j`를 선언한다.

② `for (i = 2; i <= 9; i++) {` 반복 변수 `i`가 2에서 시작하여 1씩 증가하면서 9보다 작거나 같은 동안 ③~④번을 반복하여 수행한다. 결국 ③~④번을 8번 반복하여 수행한다.

③ `for (j = 1; j <= 9; j++) {` 반복 변수 `j`가 1에서 시작하여 1씩 증가하면서 9보다 작거나 같은 동안 ④번을 반복하여 수행한다. 결국 ④번을 9번 반복하여 수행한다.

④ `System.out.printf("%d * %d = %d\n", i, j, i*j);`

결과

```
2 * 1 = 2  
2 * 2 = 4  
:  
9 * 8 = 72  
9 * 9 = 81
```

```
    }  
}  
}
```