

제어문

[1] 제어문의 개요

제어문은 서술된 순서에 따라 무조건 위에서 아래로 실행되는 프로그램의 순서를 변경할 때 사용하는 명령문이다

[2] c문제

```
#include <stdio.h>
main() {
    int score[] = { 86, 53, 95, 76, 61 };
    char grade;
    char str[] = "Rank";
    for (int i = 0; i < 5; i++) {
        switch (score[i] / 10) {
            case 10:
            case 9:
                grade = 'A';
                break;
            case 8:
                grade = 'B';
                break;
            case 7:
                grade = 'C';
                break;
            default: grade = 'F';
        }
        if (grade != 'F')
            printf( "%d is %c %s\n", i + 1, grade, str);
    }
}
```

풀이

```
#include <stdio.h>
main() {
    int score[] = { 86, 53, 95, 76, 61 }; // 5개의 요소를 갖는 정수형 배열 score를 선언
    char grade; // 문자형 변수 grade 선언
    char str[] = "Rank"; // 문자형 배열 str를 선언 'Rank'로 초기화
    for (int i = 0; i < 5; i++) { // i=0 부터 4까지 1씩 증가 i=0,1,2,3,4
        switch (score[i] / 10) { // 86/10=8 , 53/10=5, 95/10=9, 76/10=7, 61/10=6
            case 10:
            case 9: // 10, 9 이면 'A'
                grade = 'A';
                break;
            case 8: // 8 이면 'B'
```

```

        grade = 'B';
        break;
    case 7: // 7 이면 'C'
        grade = 'C';
        break;
    default: grade = 'F'; // 나머지는 'F'
    }
    if (grade != 'F') // 'F'랑 같지 않으면
        printf( "%d is %c %s\n", i + 1, grade, str);
    // %d: 10진정수, %c: 문자, %s: 문자열
}
}
결과

```

```

1 is B Rank
3 is A Rank
4 is C Rank

```

[3] Java 문제

```

public class Test {
    public static void main(String[] args) {
        String str = "agile";
        int x[] = { 1, 2, 3, 4, 5 };
        char y[] = new char[5];
        int i = 0;
        while (i < str.length()) {
            y[i] = str.charAt(i);
            i++;
        }
        for (int p : x) {
            i--;
            System.out.print(y[i]);
            System.out.print(p + " ");
        }
    }
}

```

풀이

```

public class Test {
    public static void main(String[] args) {
        String str = "agile"; // 문자열 변수 str 선언 "agile"로 초기화
        int x[] = { 1, 2, 3, 4, 5 }; // 5개의 요소를 갖는 정수 배열 x를 선언하고 초기
        char y[] = new char[5]; // 5개의 요소를 갖는 문자 배열 y를 선언
        int i = 0; // 정수형 변수 i를 선언, 0으로 초기화
        // 반복문 while(조건식) 조건식이 참일 동안 반복
        while (i < str.length()) { // i < 5 => 조건에 만족하지 않으면 반복문을 빠져
            나온다
        }
    }
}

```

```

        y[i] = str.charAt(i); // y[i]에 str의 i번째에 있는 문자를 저장
        // charAt(): 문자열에서 지정된 위치의 문자를 반환
        i++; // i를 1씩 증가, i=1, 2, 3, 4, 5
    } // y[0]="a", y[1]="g", y[2]="i", y[3]="l", y[4]="e"

    for (int p : x) { // 향상된 for 반복문, x요소가 5개라서 5번 반복
        i--; // i는 1씩 감소, i= 4,3,2,1,0
        // print() : 줄바꿈하지 않음
        System.out.print(y[i]); // y[4] y[3] y[2] y[1] y[0]
        System.out.print(p + " "); // 1 2 3 4 5
    } // e1 l2 i3 g4 a5
    }
}
결과 : e1 l2 i3 g4 a5

```

기출 따라잡기

[문제 01]

다음은 변수 n에 저장된 10진수를 2진수로 변환하여 출력하는 Java 프로그램이다
프로그램을 분석하여 괄호(1, 2)에 들어갈 알맞은 답을 쓰시오.

10을 2진수로 변환 : 1010

```

public class Test {
    public static void main(String[] args) {
        int a[] = new int[8];
        int i = 0;
        int n = 10;
        while( ( 1번 ) ) {
            a[i++] = ( 2번 );
            n /= 2;
        }
        for(i = 7; i >= 0; i- -)
            System.out.print(a[i]);
    }
}

```

풀이

10을 2진수로 변환

```

public class Test {
    public static void main(String[] args) {
        int a[] = new int[8];
        // 크기가 8인 정수형 배열 변수 a선언
        int i = 0;
        int n = 10; // 10진수 10을 n에 할당
        while(n > 0) { // n이 0보다 크면 반복

```

```

        a[i++] = n % 2; //a[0]=10%2, a[1]=5%2, ..
        n /= 2; // n = n / 2;
    }
    for(i = 7; i >= 0; i- -) // i=7,6,5,4,3,2,1,0
        System.out.print(a[i]); // a[7] a[6] a[5] a[ 4]
    // 0 0 0 0 1 0 1 0
}
}

```

[문제 02]

```

public class Test {
    public static void main(String[] args) {
        int ary[][] = new int[( 1 )][( 2 )];
        int n = 1;
        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 5; j++) {
                ary[i][j] = j * 3 + i + 1;
                System.out.print(ary[i][j] + " ");
            }
            System.out.println( );
        }
    }
}

```

풀이

```

public class Test {
    public static void main(String[] args) {
        int ary[][] = new int[3][5]; // 3행 5열 배열 선언
        int n = 1;
        for(int i = 0; i < 3; i++) { // i = 0, 1, 2
            for(int j = 0; j < 5; j++) { // j = 0, 1, 2, 3, 4
                ary[i][j] = j * 3 + i + 1; // ary[0][0]= 0*3+0+1 = 1, ...
                System.out.print(ary[i][j] + " ");
            }
            System.out.println( ); // 줄바꿈
        }
    }
}

```

결과

```

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

```

[문제 03]

```
public class Test {
    public static void main(String[] args) {
        int i = 0, c = 0;
        while (i < 10) {
            i++;
            c *= i;
        }
        System.out.println(c);
    }
}
```

풀이

```
public class Test {
    public static void main(String[] args) {
        int i = 0, c = 0;
        while (i < 10) {
            i++; // i=0,1,2,3,4,5,6,7,8,9,10
            c *= i; // c = c * i
            // 0을 계속 곱해도 0 이다
        }
        System.out.println(c); // 0
    }
}
결과 : 0
```

[문제 04]

```
public class Test{
    public static void main(String[] args){
        int a = 0, sum = 0;
        while (a < 10) {
            a++;
            if (a % 2 == 1)
                continue;
            sum += a;
        }
        System.out.println(sum);
    }
}
```

풀이

```
public class Test{
    public static void main(String[] args){
        int a = 0, sum = 0;
```

```

        while (a < 10) {
            a++; // a=1,2,3,4,5,6,7,8,9,10
            if (a % 2 == 1) // a를 2로 나누어서 나머지가 1이면 건너뛸
                continue; // 건너뛸
            sum += a; // sum = sum + a , 짝수의 누적합계
        } // 2, 4, 6, 8, 10
        System.out.println(sum); // 30
    }
}
결과 : 30

```

[문제 05]

```

#include <stdio.h>
main() {
    int c = 1;
    switch (3) {
        case 1: c += 3;
        case 2: c++;
        case 3: c = 0;
        case 4: c += 3;
        case 5: c -= 10;
        default: c--;
    }
    printf("%d", c);
}

```

풀이

```

#include <stdio.h>
main() {
    int c = 1;
    switch (3) {
        case 1: c += 3;
        case 2: c++;
        case 3: c = 0; // break문이 없어서 계속 실행
        case 4: c += 3; // c = c + 3 , c = 3
        case 5: c -= 10; // c = c - 10, c = -7
        default: c--; // c = -8
    }
    printf("%d", c); // -8
}
결과 : -8

```

[문제 06]

```
#include <stdio.h>
main() {
    int input = 101110;
    int di = 1;
    int sum = 0;
    while (1) {
        if (input == 0) break;
        sum = sum + (input % 10) * di;
        di = di * 2;
        input = input / 10;
    }
    printf("%d", sum);
}
```

풀이

```
#include <stdio.h>
main() {
    int input = 101110;
    int di = 1;
    int sum = 0;
    while (1) { // 무한반복문
        if (input == 0) break; // input = 0이면 반복문을 빠져나옴
        sum = sum + (input % 10) * di;
        di = di * 2; // id = 2, 4, 8, 16,
        input = input / 10; // 10111, 1011, 101, 10, 1, 0
    }
    printf("%d", sum); // 46
}
결과 : 46
```

[문제 07]

```
#include <stdio.h>
main() {
    int s, el = 0;
    for (int i = 6; i <= 30; i++) {
        s = 0;
        for (int j = 1; j <= i / 2; j++)
            if (i % j == 0)
                s = s + j;
        if (s == i)
            el++;
    }
    printf("%d", el);
}
```

풀이

완전수 : 약수 중 자신을 제외한 약수를 모두 합하면 자신과 같아지는 수

- 6, 28, 496, 8128,...
- $6 = 1 + 2 + 3$
- $28 = 1 + 2 + 4 + 7 + 14$

```
#include <stdio.h>
main() {
    int s, el = 0;
    for (int i = 6; i <= 30; i++) { // i=6,7,8,9,10,...
        s = 0;
        for (int j = 1; j <= i / 2; j++) // j=1, 2, 3  1,2,3 1,2,3,4 1,2,3,4
            if (i % j == 0) // 나머지가 0이면 약수
                s = s + j; // 약수의 누적합계
        if (s == i) // 정수와 약수의 누적합계가 같으면
            el++; // el 1증가
    }
    printf("%d", el);
}
결과 : 2
```

[문제 08]

```
public class Test {
    public static void main(String[] args) {
        int result[] = new int[5];
        int arr[] = { 77, 32, 10, 99, 50 };
        for(int i = 0; i < 5; i++) {
            result[i] = 1;
            for(int j = 0; j < 5; j++)
                if(arr[i] < arr[j])
                    result[i]++;
        }
        for(int k = 0; k < 5; k++)
            System.out.print(result[k]);
    }
}
```

풀이

```
public class Test {
    public static void main(String[] args) {
        int result[] = new int[5]; // 크기가 5인 정수형 배열선언
        int arr[] = { 77, 32, 10, 99, 50 }; // 배열 선언
        for(int i = 0; i < 5; i++) { // i=0,1,2,3,4
            result[i] = 1; //
```



```

        for(int j = 0; j < 5; j++) // j=0,1,2,3,4
            if(arr[i] < arr[j]) // arr[0]<arr[0]
                // 자기보다 크면 1증가
                result[i]++; // 1증가
    }
    for(int k = 0; k < 5; k++) // k=0,1,2,3,4
        System.out.print(result[k]); // result[0] result[1] ...
    }
}
결과 : 24513

```

[문제 09]

```

public class Test {
    public static void main(String[] args) {
        int r = 0;
        for (int i = 1; i < 999; i++) {
            if (i % 3 == 0 && i % 2 == 0)
                r = i;
        }
        System.out.print(r);
    }
}

```

풀이

```

public class Test {
    public static void main(String[] args) {
        int r = 0;
        for (int i = 1; i < 999; i++) { // i=1,2,3,4,5,... 998
            if (i % 3 == 0 && i % 2 == 0)
                // 3과 2로 나누어서 나머지가 0
                r = i;
        }
        System.out.print(r);
    }
    // 1부터 할수 없다 998부터 디버깅한다
}
결과 : 996

```

[문제 10]

```

public class Test {
    public static void main(String[] args) {
        int j, i;
        for (j = 0, i = 0; i <= 5; i++) {
            j += i;
        }
    }
}

```

```

        System.out.print(i);
        if (i == 5) {
            System.out.print("=");
            System.out.print(j);
        }
        else
            System.out.print("+");
    }
}
}

```

풀이

```

public class Test {
    public static void main(String[] args) {
        int j, i;
        for (j = 0, i = 0; i <= 5; i++) {    j=0 i=0 1 2 3 4 5
            j += i; // j = j + i
            System.out.print(i); 0
            if (i == 5) {
                System.out.print("=");
                System.out.print(j);
            }
            else
                System.out.print("+"); // +
        }
    }
}
결과 : 0+1+2+3+4+5=15

```

[문제 11]

```

#include <stdio.h>
main() {
    int n[] = { 5, 4, 3, 2, 1 };
    for (int i = 0; i < 5; i++)
        printf("%d", n[(i + 1) % 5]);
}

```

풀이

```

#include <stdio.h>
main() {
    int n[] = { 5, 4, 3, 2, 1 }; // n 배열 생성
    for (int i = 0; i < 5; i++) // i = 0 1 2 3 4
        printf("%d", n[(i + 1) % 5]);
    n[(1)%5=1] ...
}

```

```
}
결과 : 43215
```

[문제 12]

```
public class Test {
    public static void main(String[] args) {
        int aa[][] = { {45, 50, 75},{89} };
        System.out.println(aa[0].length);
        System.out.println(aa[1].length);
        System.out.println(aa[0][0]);
        System.out.println(aa[0][1]);
        System.out.println(aa[1][0]);
    }
}
```

풀이

```
public class Test {
    public static void main(String[] args) {
        int aa[][] = { {45, 50, 75},{89} }; // 2차원 배열 선언
        System.out.println(aa[0].length); // 0
        System.out.println(aa[1].length); // 1
        System.out.println(aa[0][0]); // 45
        System.out.println(aa[0][1]); // 50
        System.out.println(aa[1][0]); // 89
    }
}
```

결과

```
3
1
45
50
89
```

[문제 13]

정수를 역순으로 출력, 1234의 역순은 4321
1230 처럼 0으로 끝나는 정수는 고려하지 않는다

```
#include <stdio.h>
int main() {
    int number = 1234;
    int div = 10, result = 0;
    while (number ( 1 ) 0) {
        result = result * div;
        result = result + number ( 2 ) div;
```

```

        number = number ( (3) ) div;
    }
    printf("%d", result);
}

```

풀이

```

#include <stdio.h>
int main() {
    int number = 1234;
    int div = 10, result = 0;
    while (number ( (1) ) 0) { // number 가 0과 비교하여 false 이면 빠져나옴
        result = result * div;
        result = result + number ( (2) ) div;
        number = number ( (3) ) div;
    }
    printf("%d", result); // result : 역순으로 나올 변수
}

```

```

#include <stdio.h>
int main() {
    int number = 1234;
    int div = 10, result = 0;
    while (number != 0) { // number 값이 0이면 반복문을 빠져나온다
        result = result * div; // 0,
        result = result + number % div; // 4
        number = number / div;
    }
    printf("%d", result);
}
결과 : 4321

```