

상명튜터링

2 주 차 조 건 문

조건문 if~else if~else문 작동 원리

```
if(조건식){  
  //작동 코드  
}  
else if(if 조건을 제외한 조건식){  
  //작동 코드  
}  
else{  
  //if와 else if 조건을 제외한 모든 경우의 수  
  //작동 코드  
}
```

제약 조건

if와 else는 1개만 작성 가능하다.

else if문은 원하는 만큼 작성이 가능하다.

조 건 문 if문을 중심으로

```
#include <stdio.h>

int main() {
    int number_A = 6500;

    if (number_A > 5000){
        printf("number_A가 %d으로 5000 보다 크다!!\n", number_A);
    }
    else if(number_A < 5000){
        printf("number_A가 %d으로 5000 보다 작다!!\n\n", number_A);
    }
    else {
        printf("number_A가 %d으로 5000이다!!\n\n", number_A);
    }
    return 0;
}
```

```
./main
number_A가 6500으로 5000 보다 크다!!
```

< 결 과 >

number_A가 if 조건문의 조건식
number_A > 5000을 만족하므로 if
조건문의 작동코드가 실행되게 된다.

조 건 문 else if문을 중심으로

```
#include <stdio.h>
```

```
int main() {  
    int number_A = 4500;  
  
    if (number_A > 5000){  
        printf("number_A가 %d으로 5000 보다 크다!!\n", number_A);  
    }  
    else if(number_A < 5000){  
        printf("number_A가 %d으로 5000 보다 작다!!\n\n", number_A);  
    }  
    else {  
        printf("number_A가 %d으로 5000이다!!\n\n", number_A);  
    }  
    return 0;  
}
```

```
./main  
number_A가 4500으로 5000 보다 작다!!
```

< 결 과 >

number_A가 else if 조건문의 조건식
number_A < 5000을 만족하므로 else if
조건문의 작동코드가 실행되게 된다.

조 건 문 else문을 중심으로

```
#include <stdio.h>

int main() {
    int number_A = 5000;

    if (number_A > 5000){
        printf("number_A가 %d으로 5000 보다 크다!!\n", number_A);
    }
    else if(number_A < 5000){
        printf("number_A가 %d으로 5000 보다 작다!!\n\n", number_A);
    }
    else {
        printf("number_A가 %d으로 5000이다!!\n\n", number_A);
    }
    return 0;
}
```

```
➤ ./main
number_A가 5000으로 5000이다!!
```

< 결 과 >

number_A가 if와 else if 조건식을 둘 다 만족하지 않으므로 else의 작동코드가 실행 되 게 된 다 .

조건문 switch~case~default문의구조

```
switch(case를 판단할 변수){  
    case 조건식:  
        //작동 코드  
        break;  
    default:  
        //위의 case에서 처리하지 않은 모든 경우의 수  
        //작동 코드  
        break;  
}
```

제약 조건

default는 1개만 작성 가능하다.

case는 원하는 만큼 작성이 가능하다.

작동 코드를 작성한 후에는 break; 를 작성해야한다.

조건문 switch~case~default문 예시

```
#include <stdio.h>
```

```
int main() {  
    for (int i = 0; i < 7; i++) {  
        switch(i){  
            case 0:  
            case 1:  
                printf("1. i의 숫자가 %d이다.\n", i);  
                break;  
            case 2:  
                printf("2. i의 숫자가 %d이다.\n", i);  
                break;  
            case 3:  
            case 4:  
            case 5:  
                if(i == 3){  
                    printf("3-1. i의 숫자가 %d이다.\n",  
                        i);  
                }  
            }  
        }  
    }  
}
```

```
        else if(i == 4){  
            printf("3-2. i의 숫자가 %d이다.\n", i);  
        }  
        else{  
            printf("3-3. i의 숫자가 %d이다.\n", i);  
        }  
        break;  
    default:  
        printf("4. i의 숫자가 %d으로 case 조건 범위  
            밖이다.\n", i);  
        break;  
    }  
}  
return 0;  
}
```

조 건 문 switch~case~default문 예시 결과

```
1. i의 숫자가 0이다.  
1. i의 숫자가 10이다.  
2. i의 숫자가 20이다.  
3-1. i의 숫자가 30이다.  
3-2. i의 숫자가 40이다.  
3-3. i의 숫자가 50이다.  
4. i의 숫자가 6으로 case 조건식 범위 밖이다.  
▶
```

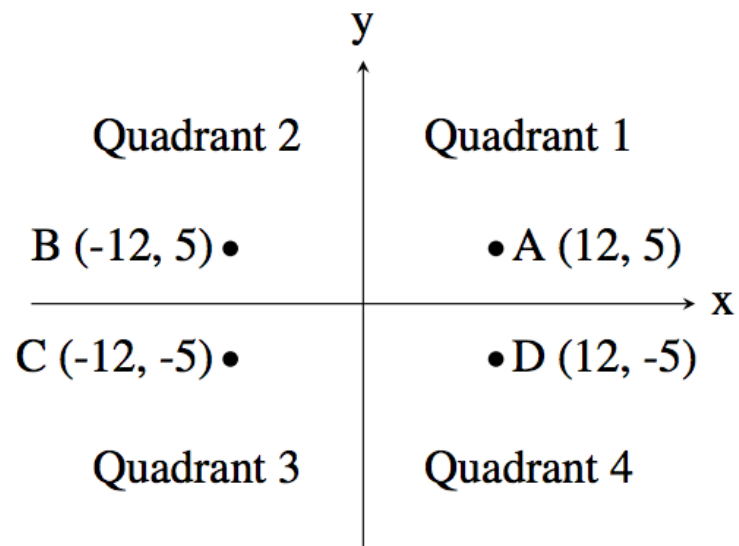
< 결 과 >

break;를 사용하지 않으면 다음 case의 코드까지 실행되게 된다.
따라서 0과 1은 printf(1.~)을 출력하고, 2는 printf(2.~)를 출력한다.

3,4,5를 통해서 switch~case문 안에 if문을 작성할 수 있다는 것을 알 수 있다.
따라서 3은 printf(3-1.~), 4는 printf(3-2.~), 5는 printf(3-3.~)를
출력한다.

마지막으로 default는 이전 case에서 처리해주지 않은 경우의 수를
출력해준다.
따라서 6은 이전 case에서 처리하지 못하였으므로 printf(4.~)을 출력한다.

숙제



<몇 사분면에 속하는 지 출력하기>

문제해설

점의 좌표를 입력받아 그 점이 어느 사분면에 속하는지 알아내는 프로그램을 작성하시오.
단, x좌표와 y좌표는 모두 양수나 음수라고 가정한다.

결과해설

제 1,2,3,4 사분면 예시를 각각 하나씩 만들어서 출력한 결과를 캡처하여 보내시오.

```
00의 숙제  
2개의 정수를 공백을 기준으로 입력해주세요 >>10 12  
x=10, y=12이므로 제 1사분면에 속한다. >
```

```
00의 숙제  
2개의 정수를 공백을 기준으로 입력해주세요 >>-5 20  
x=-5, y=20이므로 제 2사분면에 속한다. >
```

```
00의 숙제  
2개의 정수를 공백을 기준으로 입력해주세요 >>-10 -15  
x=-10, y=-15이므로 제 3사분면에 속한다. >
```

```
00의 숙제  
2개의 정수를 공백을 기준으로 입력해주세요 >>2000 -2500  
x=2000, y=-2500이므로 제 4사분면에 속한다. >
```

<결과 예시>