

Introduction to Computer and Lab

Homework #3

Due date: Apr 14, 2016

학번: 201404051

이름: 정 용 석

1. 배수 여부

1.1 Solution

이 함수 같은 경우는 생각보다 간단하다. 두 수 A, B 가 서로에게 배수이다 라는 말은 즉 A 를 B 로 나누거나, B 를 A 로 나누었을 때 나누어 떨어진다는 말이기 때문에, % mod 를 이용하여 간단하게 조건 문을 만들 수 있었다. 하지만 여기서 1 가지 예외는 바로 A 혹은 B 가 0 일 때이다. 두 수중 하나라도 0 이 된다면 수학적으로 어떤 수를 0 으로 나누는 것은 불가능하다. 따라서 입력이 0 이 들어온다면 프로그램이 에러를 일으키고, 중단된다. 이를 처리하기 위한 예외 문을 작성하여야 하지만, 현재까지 수업 내용으로는 불가능하다고 판단, 유저 인터페이스로 이런 문제를 피하도록 경고문을 보여주는 것으로 끝냈다.

1.2 Source Code

```
//사용자에게 A,B 두 수를 입력 받고 A가 B의 배수, 혹은 B가 A의 배수인지 판별
void checkCommonDivisor() {
    int A, B;
    printf("Insert numbers A and B(A nor B cannot be 0): ");
    fflush(stdout);
    scanf("%d %d", &A, &B);

    //A가 B로 나누어지거나, B가 A로 나누어지면 서로의 배수
    if(A % B == 0 || B % A == 0)
        printf("YES!\n");
    else
        printf("NO!\n");
}
```

1.3 Result (Snap Shot)

```
why.exe [C/C++ Application] C:\home\why\Debug\why.exe (4/9/16, 5:23 PM)
Insert numbers A and B(A nor B cannot be 0): 12 36
YES!

Insert numbers A and B(A nor B cannot be 0): 15 32
NO!

Insert numbers A and B(A nor B cannot be 0): 158 789
NO!

Insert numbers A and B(A nor B cannot be 0): 204 2
YES!

Insert numbers A and B(A nor B cannot be 0): 13 13
YES!
```

2. 삼각형 가능 여부

2.1 Solution

세 변의 길이를 사용자에게 입력 받았을 때, 임의의 두 변의 길이가 다른 한 변의 길이보다 모든 경우에서 더 커야 삼각형이 된다. 따라서 두 변의 길이를 이루는 경우의 수는 $A+B$, $B+C$, 그리고 $A+C$ 이 3 가지 밖에 없기에 이 모두를 만족하기 위한 $\&\&$ 연산을 이용하여 조건 문을 작성하였다.

2.2 Source Code

```
//사용자에게 세 변의 길이를 입력받고, 삼각형의 가능 여부를 판단
void checkTriangle() {

    int sideA, sideB, sideC;
    printf("Insert three sides A, B and C: ");
    fflush(stdout);
    scanf("%d %d %d", &sideA, &sideB, &sideC);

    //임의의 두변의 길이가 다른 한변보다 모든 경우에서 더 커야 삼각형이다.
    if(sideA < sideB+sideC && sideB < sideA+sideC && sideC <
sideA+sideB)
        printf("YES!\n");
    else
        printf("NO!\n");
}
```

2.3 Result (Snap Shot)

why.exe [C/C++ Application] C:\home\why\Debug\why.exe (4/9/16, 5:31 PM)

Insert three sides A, B and C: 1 2 4

NO!

Insert three sides A, B and C: 3 1 6

NO!

Insert three sides A, B and C: 20 20 20

YES!

Insert three sides A, B and C: 30 30 30

YES!

Insert three sides A, B and C: 50 100 2

NO!

3. 자유 이용권 계산

3.1 Solution

일단 조건 문을 크게 두 가지 경우로 나누었다. 처음을 나이로 나누고 그리고 시간으로 나누어서 조건 문에 조건 문을 추가하는 형태를 사용했다. 즉, 사용자에게 입력 받은 정보를 이용하여 처음에는 대인과 소인으로 나누고, 대인일 때와 소인일 때 각각 시간 별 가격을 출력하게 해주었다. 이도 배수 문제와 비슷하게, 시간과 분은 각각 0~24, 그리고 0~60 까지 수 밖에는 가질 수 없다. 따라서 이를 초과하거나 음수의 수를 입력 받았을 때의 예외 문은 따로 추가하지 않았다.

3.2 Source Code

```
void calculateFee() //사용자에게 현재 시간과 나이를 입력받고 자유이용권 이용 요금 출력
{
    int hour, min, age;
    printf("Insert time(hour and minute(s)), and your age: ");
    fflush(stdout);
    scanf("%d %d %d", &hour, &min, &age);

    // 12세 초과 65세 미만일 경우
    if(age > 12 && age < 65)
        //오전9시 ~ 오후 5시
        if(hour > 9 && min >= 0 && hour < 18 && min < 60)
            printf("34000 won\n");
        //그 이외의 시간
        else
            printf("10000 won\n");
    // 12세 이하 65세 이상일 경우
    else
        //오전9시 ~ 오후 5시
        if(hour > 9 && min >= 0 && hour < 18 && min < 60)
            printf("25000 won\n");
        //그 이외의 시간
        else
            printf("10000 won\n");
}
```

3.3 Result (Snap Shot)

```
why.exe [C/C++ Application] C:\home\why\Debug\why.exe (4/9/16, 5:42 PM)
Insert time(hour and minute(s)), and your age: 12 33 34
34000 won

Insert time(hour and minute(s)), and your age: 8 10 1
10000 won

Insert time(hour and minute(s)), and your age: 18 30 24
10000 won

Insert time(hour and minute(s)), and your age: 15 00 24
34000 won

Insert time(hour and minute(s)), and your age: 24 00 72
10000 won

Insert time(hour and minute(s)), and your age: 15 00 30
34000 won

Insert time(hour and minute(s)), and your age: 14 0 9
25000 won
```

4. 근로소득세액 계산

4.1 Solution

일단은 const 상수를 이용하여 과세 표준 구간 별 가격을 입력하였다. 이를 사용하지 않고 코드 작성 시, 모양도 이상하고, 읽기에도 벅차기에 따로 선언과 초기화를 했다. 이 후, 사용자에게 입력 받은 근로소득세를 조건 문을 이용하여 알맞은 구간에 입력 받도록 작성하였다. 그리고 각 구간 별 계산법을 적용하여 합을 구하고 저장하여 이를 출력하도록 하였다. 입력 받은 수가 수십억 규모의 수일 수도 있기에 long long 자료형을 이용하여 입, 출력을 진행하였다. 하지만 문제점은 컴파일러에 있었다. 저번 과제에서도 long long 자료형을 이용한 입, 출력 시에 컴파일러가 이를 인식하지 못하는 문제점이 있었다. 마찬가지로 동일한 반응을 보여주었고, 1 억이상의 수에서는 차이점을 보이지 않았지만, 그 이하의 수에서는 출력 값이 조금 다르게

나오는 예러가 있었다. 예를 들면 1 천만원의 소득세를 입력했을 때, 예상 값은 6 십만원이 나와야 하지만, 현재 내 컴퓨터의 이클립스로 컴파일, 실행 시 599999 의 수가 나온다. 이는 1 억 이하의 모든 경우에 해당된다. 처음에는 자료 형 자동 변환 간에 데이터 손실에 의한 것이라고 판단, 여러 다양한 방법을 시도했지만, 결과는 똑같았다. 비교를 위하여 Visual Studio 에서 프로그램을 실행시켰을 때는, 아무 문제없이 6 십만이라는 값을 출력해주었다. 따라서 나는 이를 컴파일 버전과 관련이 있다고 판단하고, 코딩에는 아무 이상이 없다고 확신한다.

4.2 Source Code

```
void calculateTax() //사용자에게 근로소득세를 입력 받고, 근로소득세액을 출력
{
    long long income;
    long long total;
    printf("Enter your income: ");
    fflush(stdout);
    scanf("%lld" , &income);

    const int sec1 = 12000000;    //과세 표준1 1200만원
    const int sec2 = 46000000;    //과세 표준2 4600만원
    const int sec3 = 88000000;    //과세 표준3 8800만원
    const int sec4 = 150000000;   //과세 표준4 1억5000만원

    if(income <= sec1) //과세 표준 1200만원 이하
        total=income*(0.06);
    else if(income <= sec2) //과세 표준 4600만원 이하
        total = (sec1 * 0.06) + ((income-sec1) * 0.15);
    else if(income <= sec3) //과세 표준 8800만원 이하
        total = (sec1 * 0.06) + ((sec2-sec1) * 0.15) + ((income-
sec2) * 0.24);
    else if(income <= sec4) //과세 표준 1억5000만원 이하
        total = (sec1 * 0.06) + ((sec2-sec1) * 0.15) + ((sec3-sec2)
* 0.24) + ((income-sec3) * 0.35);
    else if(income > sec4) //과세 표준 1억 5000만원 초과
        total = (sec1 * 0.06) + ((sec2-sec1) * 0.15) + ((sec3-sec2)
* 0.24) + ((sec4-sec3) * 0.35) + ((income-sec4) * 0.38);

    printf("Total: %lld\n", total);
}
```

4.3 Result (Snap Shot)

<Visual Studio Result>

```
C:\> C:\WINDOWS\system32\cmd.exe
Enter your income: 1000000
INCOME: 1000000
Total: 60000

Enter your income: 12345678
INCOME: 12345678
Total: 771851

Enter your income: 432115678918
INCOME: 432115678918
Total: 164184557988

Enter your income: 43215678910
INCOME: 43215678910
Total: 16402557985

Enter your income: 500111111
INCOME: 500111111
Total: 170642222
```

<Eclipse Result>

```
why.exe [C/C++ Application] C:\home\why\Debug\why.exe (4
Enter your income: 10000000
Total: 599999

Enter your income: 12345678
Total: 771851

Enter your income: 432115678918
Total: 164184557988

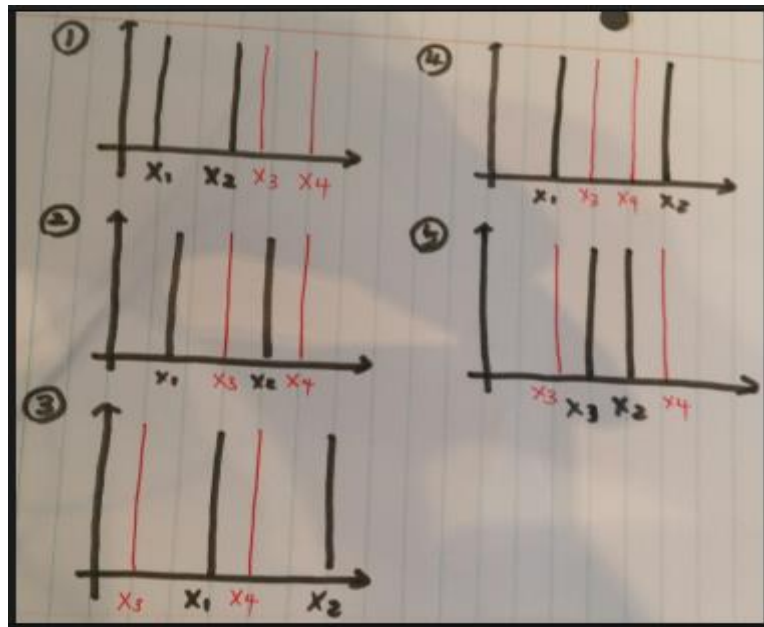
Enter your income: 43215678910
Total: 16402557985

Enter your income: 500111111
Total: 170642222
```

5. 사각형의 겹치는 넓이 계산

5.1 Solution

이 문제는 경우의 수가 굉장히 다양해 보이지만 사실상 그렇지 않다. 내가 한 방법은 다음과 같다. 밑변을 width, 높이를 height 으로 두고, 두 변수를 따로따로 구하는 방식을 선택하였다. 두개의 사각형의 x 좌표들이 겹칠 수 있는 경우의 수는 밑의 그림과 같다.



이를 바탕으로 각 조건 문을 작성하였고, 각 상황에 따라 밑변을 구하는 형식으로 코드를 작성하였다. Height, 높이도 마찬가지로 같은 경우의 수를 가진다. 따라서, x 를 이용한 조건 문과 y 를 이용한 조건 문은 변수만 다르지 조건은 모두 동일하다. 하지만 한 가지 전제 조건 하에 위의 조건들을 사용할 수 있다. $x_1 \leq x_2$, 그리고 $x_3 \leq x_4$ 의 조건이다. 이는 문제 내용에 서술되어 있으므로, 따로 생각하진 않았다. 이렇게 구한 width 와 height 를 이용하여 겹친 사각형의 넓이, area 를 구하고 이를 출력하였다.

5.2 Source Code

```
void calculateOverlapRect() //사용자에게 두 개의 사각형의 좌표를 입력 받고, 겹치는
넓이 출력
{
    int x1,x2,x3,x4;
    int y1,y2,y3,y4;
    int width, height;
    int area;

    /* 참고로 이 프로그램은 x1<=x2, x3<=x4라는 전제조건이 깔려있다. */
    printf("Insert two rectangles: ");
    fflush(stdout);
    scanf("%d %d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3, &x4,
&y4);

    if(x2 <= x3) //1번째 사각형과 2번째 사각형이 겹치지 않을때
        width = 0;
    else if(x1 < x3 && x3 < x2 && x2 <= x4) //1번째 사각형이 2번째 사각형 왼쪽에
    겹칠 때
        width = x2 - x3;
    else if(x3 <= x1 && x1 < x4 && x4 < x2) //1번째 사각형이 2번째 사각형
    오른쪽에 겹칠 때
        width = x4 - x1;
    else if(x1 <= x3 && x4 <= x2) //1번째 사각형 안에 2번째 사각형이
    겹칠 때
        width = x4 - x3;
    else if(x3 <= x1 && x2 <= x4) //1번째 사각형이 2번째 사각형 안에
    겹쳐질 때
        width = x2 - x1;

    //이하동문
    if(y2 <= y3)
        height = 0;
    else if(y1 < y3 && y3 < y2 && y2 <= y4)
        height = y2 - y3;
    else if(y3 <= y1 && y1 < y4 && y4 < y2)
        height = y4 - y1;
    else if(y1 <= y3 && y4 <= y2)
        height = y4 - y3;
    else if(y3 <= y1 && y2 <= y4)
        height = y2 - y1;

    area = width*height;
    printf("width: %d\n", width);
    printf("height: %d\n", height);
    printf("Area: %d\n", area);
}
```

5.3 Result (Snap Shot)

why.exe [C/C++ Application] C:\home\why\Debug\why.exe (4/9/16, 6:12 PM)

Insert two rectangles: 0 0 4 4 2 2 8 8

width: 2

height: 2

Area: 4

Insert two rectangles: 1 1 3 3 6 6 20 20

width: 0

height: 0

Area: 0

Insert two rectangles: 1 2 3 4 5 6 7 8

width: 0

height: 0

Area: 0

Insert two rectangles: 3 3 20 20 4 0 8 14

width: 4

height: 11

Area: 44

Insert two rectangles: 1 3 6 7 2 4 9 10

width: 4

height: 3

Area: 12