**Introduction to Computer and Lab**

**Homework #1**

**Due date: Mar 31. 2016**

**학번: 201404051**

**이름: 정 용 석**

**1 (6 points). PrintReport**

**1.1 Solution**

3명의 학생의 3 과목에 대한 점수를 각각 입력 받아야 하므로, 일단 각 과목별로 3개의int형 변수를 선언하였다. 그리고 학생 당 3 과목에 대한 합을 구할 int형 변수 1개와 평균을 구할 float형 변수를 1개를 추가하였다. 그리고 3과목의 점수를 한 번에 입력 받아서 각각 알맞은 변수에 저장하였다. 그런 후에는 총합을 저장하는 변수에 각 과목의 점수를 모두 합하는 연산을 수행하는 식으로 총합을 저장하고, 평균은 저장된 총합을 3과목, 하지만 여기서 그냥 3이 아닌 3.0을 나누기 연산으로 작성하여 평균의 자료 형 float에 알맞게 저장할 수 있도록 하였다.

총합과 평균을 구함과 동시에 바로바로 그 값을 출력하는 방식으로 굳이 총합을 저장하는 변수와 평균을 저장하는 변수를 여러 개 선언하지 않고, 한 개에서 그 값을 매번 연산, 초기화하는 방식을 선택했다. 출력 과정에서는 각각의 정보를 tab크기만큼 띄어쓰기를 하여서 보기 훨씬 쉬운 방향으로 작성을 하였다. 또한, 평균은 소수점 2번 자리 까지만 출력하도록 설정하였다.

**1.2. Source code**

**void** **printReport**() // 3명의 세 과목의 점수 입력 받고 점수와 총점, 평균을 출력

{

**int** math1, math2, math3;//수학

**int** eng1, eng2, eng3; //영어

**int** kor1, kor2, kor3; //국어

**int** sum; //총합

**float** average; //평균

//각가의 학생의 세 과목 점수를 한 번에 입력받는다.

**printf**("첫 번째 학생의 세 과목 점수를 입력하시오: ");

**fflush**(stdout);

**scanf**("%d %d %d", &math1, &eng1, &kor1);

**printf**("두 번째 학생의 세 과목 점수를 입력하시오: ");

**fflush**(stdout);

**scanf**("%d %d %d", &math2, &eng2, &kor2);

**printf**("세 번째 학생의 세 과목 점수를 입력하시오: ");

**fflush**(stdout);

**scanf**("%d %d %d", &math3, &eng3, &kor3);

**printf**("국어\t영어\t수학\t총점\t평균\n");

//sum과 average룰 구하고 바로 출력을 한다. 그 다음 정보를 이용하여 sum과 average 다시 초기화한다.

sum = math1 + kor1 + eng1;

average = sum / 3.0;

**printf**("%d\t%d\t%d\t%d\t%.2f\n", kor1, eng1, math1, sum, average);

sum = math2 + kor2 + eng2;

average = sum / 3.0;

**printf**("%d\t%d\t%d\t%d\t%.2f\n", kor2, eng2, math2, sum, average);

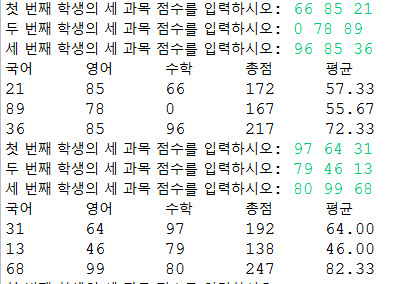
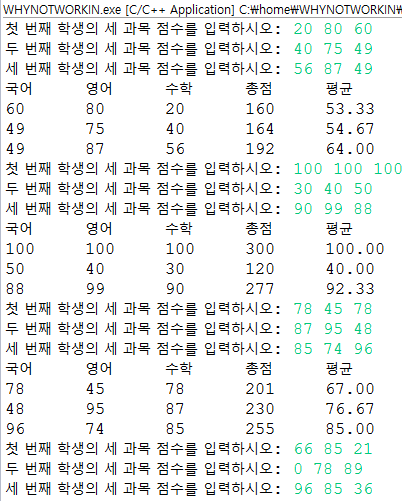
sum = math3 + kor3 + eng3;

average = sum / 3.0;

**printf**("%d\t%d\t%d\t%d\t%.2f\n", kor3, eng3, math3, sum, average);

}

**1.3. Result (snapshot)**

****

**2 (6 points). arithmetic Operations**

**2.1. Solution**

두개의 정수 x와 y의 입력 범위가 int형의 범위를 넘어가기 때문에 long long으로 변수를 선언하였다. 동시에 이 둘을 이용하여 사칙 연산 시 당연히 int형 범위를 넘어가기에 마찬가지로 long long변수로 사칙연산의 값을 저장할 수 있는 변수를 선언해주었다. 처음에는 이 변수들을 선언하지 않고, 두 개의 정수를 받고 연산한 것을 바로 출력하는 방식을 사용하였다. 하지만 프로그램 작동에는 아무 문제가 없었지만 컴파일 시 전달 인자 수와 관련 된 경고 창이 계속 생겨서 디버깅을 진행해봤지만, 확실한 답은 찾을 수가 없었고, 단지 너무 큰 자료 형의 연산과 동시에 출력에 있어서 어떤 문제가 생기는 것 같다는 결론을 내렸고, 각 사칙연산의 값을 저장하는 변수를 선언하고 값을 저장하는 방식을 선택하고 테스트 해도 경고 창들이 사라지지 않았다. 오랜 구글링 후 내릴 수 있던 결론은 컴파일러의 버전 문제인 걸로 생각된다. 하지만 프로그램 실행에 있어서는 아무 문제를 보이지 않았다.

**2.2. Source code**

**void** **arithmeticOperations**() // 입력 받은 두 수의 사칙연산 결과 출력

{

**long** **long** x, y, sum, sub, mult, div;

**printf**("두 개의 정수를 입력하시오: ");

**fflush**(stdout);

**scanf**("%lld %lld" , &x, &y);

//각각의 연산 먼저 실행 후 출력

sum = x + y;

sub = x - y;

mult = x \* y;

div = x / y;

**printf**("%lld + %lld = %lld\n", x, y, sum);

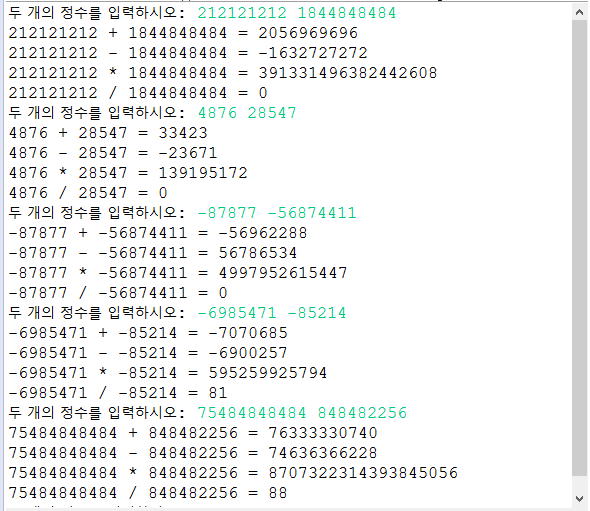
**printf**("%lld - %lld = %lld\n", x, y, sub);

**printf**("%lld \* %lld = %lld\n", x, y, mult);

**printf**("%lld / %lld = %lld\n", x, y, div);

}

**2.3. Result (snapshot)**

****

**3 (6 points). transformSpace**

**3.1. Solution**

평과 제곱 미터 값은 정수로 받는 예제 였지만, 결과 값은 소수점 두 자리 수까지 출력하는 float형의 변수이다. 사실 평과 제곱 미터 또한 소수점 자리로 받아도 상관은 없다. 일단 평을 일컫는 flat과 제곱 미터를 표현할 area 변수를 float형으로 선언하였기 때문이다. 연산은 굉장히 간단하였다. 1평은 3.3m^2 이기에 입력 받은 평의 값을 3.3으로 곱하면 제곱 미터가 되고, 입력 받은 제곱 미터의 값을 3.3으로 나누면 바로 평의 값이 된다. 출력할 시에는 입력 받은 값은 소수점을 모두 버리고, 출력해야 될 값들은 소수점 2자리까지 표현되게 작성하였다

**3.2. Source code**

**void** **transformSpace**() //평을 제곱미터로, 제곱미터를 평으로 변환하여 출력

{

**float** flat;

**float** area;

**printf**("평을 입력하시오: ");

**fflush**(stdout);

**scanf**("%f" , &flat);

area = flat \* (3.3);

**printf**("입력한 평수 %.0f 은 %.2f 제곱미터입니다\n", flat, area);

**printf**("제곱미터를 입력하시오: ");

**fflush**(stdout);

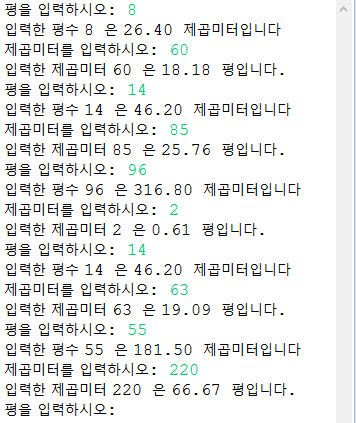
**scanf**("%f", &area);

flat = area / 3.3;

**printf**("입력한 제곱미터 %.0f 은 %.2f 평입니다.\n", area, flat);

}

**3.3. Result (snapshot)**

****

**4 (6 points). transformHeight**

**4.1. Solution**

평수 제곱 미터 문제와 비슷하게 입력 받은 연산하여 값을 출력하는 문제이다. 일단 키와 피트를 저장하는 변수는 int형으로 하였다. 키는 소수점까지 받지 않는 다는 가정하에 int형을 선언하였고, 출력 값을 보면 피트와 인치로 나뉘는데, 피트는 정수이고 인치는 소수점으로 표현될 것이다. 일단 입력 받은 키의 값을 계산하여 인치를 계산하였다. 그리고 말했듯이 피트는 정수 형으로 출력이 될 것이기 때문에 단순히 계산된 인치 값을 12로 나누어서 나머지 값을 버린 몫만 가지게 저장한 뒤, 나머지 값을 총 인치 값에서 계산된 피트 값의 인치만큼 빼서 나머지 값을 인치 변수에 다시 초기화 하였다. 그리고 저장된 값들을 피트와 소수점 세자리까지의 인치로 출력함으로서 프로그램을 마무리했다.

**4.2. Source code**

**void** **transformHeight**() // 입력받은 키를 피트로 변환하여 출력

{

**int** cm, ft;

**float** inch;

**printf**("키<cm>를 입력하시오: ");

**fflush**(stdout);

**scanf**("%d", &cm);

inch = cm / 2.54;

ft = inch / 12;

inch = inch-(12\*ft);

**printf**("입력하신 키 %d는 %d 피트 %.3f 인치입니다 \n", cm, ft, inch);

}

**4.3. Result (snapshot)**

****

**5 (6 points). printDigits**

**4.1. Solution**

일단 문제 해결에 앞서서 수학적으로 일, 십 백의 자리가 어떻게 표현되는지 생각해보았다. 예를 들어, 일, 십, 백의 자리가 각각 x, y, z일 때 이 수 f는 수학적으로 f = 1\*x+10\*y+(10^2)\*z 가 된다. 따라서 x를 구하려면, f = x+10(y + 10z)가 되는데, 이 말인 즉 슨, 입력 받은 정수를 10으로 나눈 나머지가 된다. 마찬가지로 y는 10으로 나누게 되면 f = x/10 + y + 10z가 되는데, 이는 y + (x/10 + 10z)가 되고, x의 값은 소수로서 버려지고 10으로 나눈 나머지 정수 값이 십의 자리가 된다. 백의 자리의 경우는 단순하게 100으로 나눈 몫이 바로 백의 자리가 된다.

**4.2. Source code**

**void** **printDigits**() // 1000 미만의 수 입력받아 일, 십, 백의자리 출력

{

**int** num;

**printf**("1000 미만의 정수를 입력하시오: ");

**fflush**(stdout);

**scanf**("%d", &num);

**int** one = num % 10; //일의 자리 구하기

**int** ten = num / 10 % 10;//십의 자리 구하기

**int** hundred = num / 100;//백의 자리 구하기

**printf**("%d의 백의 자리는 %d입니다\n", num, hundred);

**printf**("%d의 십의 자리는 %d입니다\n", num, ten);

**printf**("%d의 일의 자리는 %d입니다\n", num, one);

}

**4.3. Result (snapshot)**

****