**과제1 변수와 표준입출력**



|  |  |
| --- | --- |
| **과 목 :** | **기초컴퓨터프로그래밍** |
| **제출 일자 :** | **2018. 4. 09** |
| **담당 교수 :** | **김승태** |
| **학 과 :** | **사회복지학과** |
| **학 번 :** | **20170299** |
| **이 름 :** | **김유빈** |

**<본 과제 – 사칙연산>**

해결방안: 사용자는 3자리 이하의 정수를 입력하므로 변수(a, b)의 자료형을 int형으로 한다. 연산의 결과값도 정수 부분만 나오는 것으로 정했기 때문에 연산 값의 변수(total, minus, multi, divi)의 자료형도 int형으로 한다(정수끼리의 연산은 정수가 결과값으로 나오므로 int형으로 하는 것이 옳다). 표준 입출력에서 int형 정수의 서식문자로 %d를 쓴다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main()

{

int a, b;

printf("정수 두 개를 입력하세요 : ");

scanf("%d %d", &a, &b);

int total = a + b;

int minus = a - b;

int multi = a \* b;

int divi = a / b;

printf("%d + %d = %d\n", a, b, total);

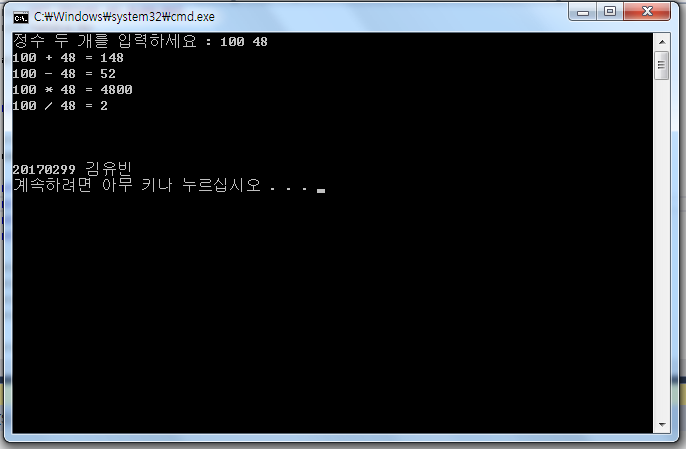
printf("%d - %d = %d\n", a, b, minus);

printf("%d \* %d = %d\n", a, b, multi);

printf("%d / %d = %d\n\n\n\n", a, b, divi);

printf("20170299 김유빈\n");

}



**<본 과제 – 단리 복리>**

해결방안: 단리는 원금에 대해서만 이자가 붙는 것이고, 복리는 원금 + 이자에 대하여 이자가 붙는 것이다. 둘 다 매 년 더해지거나 곱해지는 값이 반복되므로 복합 대입 연산자를 이용하여 구한다. 1년의 단리와 복리를 정의하고 2년부터는 복합 대입 연산자를 이용한다. 단리는 매해 원금\*(금리\*0.01) 즉 이자가 더해지고 복리는 매해 (1+금리\*0.01)가 곱해진다. 따라서 단리에는 +=를, 복리에는 \*=를 사용한 다음 printf 로 출력하면 다음해의 단리와 복리가 계산된다.

참고로 1번째 simple의 수식을 보면 원금(money)과 금리(r)가 정수형 int여도 r에 0.01(실수)를 곱하여 double형 계산이 행해진다는 점이다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main()

{

int money; // 원금의 변수는 money

int r; // 금리의 변수는 r(원금과 금리는 정수이므로 정수형 int사용)

printf("원금을 입력하세요(원). ");

scanf("%d", &money);

printf("금리를 입력하세요(%c). ", 37); //문자기호'%"를 출력하기 위해

서식문자 '%c'를 이용

scanf("%d", &r);

printf("원금은 %d원 금리 %d%c 입니다. \n", money, r, 37);

double simple, compound; //단리는 simple, 복리는 compound

simple = money + money \* (r\*0.01);

compound = money\*(1 + r \* 0.01);

printf("기간 단리 복리\n");

printf("1년 %.2f %.2f\n", simple, compound);

simple += money \* (r\*0.01);

compound \*= 1 + r \* 0.01;

printf("2년 %.2f %.2f\n", simple, compound);

simple += money \* (r\*0.01);

compound \*= 1 + r \* 0.01;

printf("3년 %.2f %.2f\n", simple, compound);

simple += money \* (r\*0.01);

compound \*= 1 + r \* 0.01;

printf("4년 %.2f %.2f\n", simple, compound);

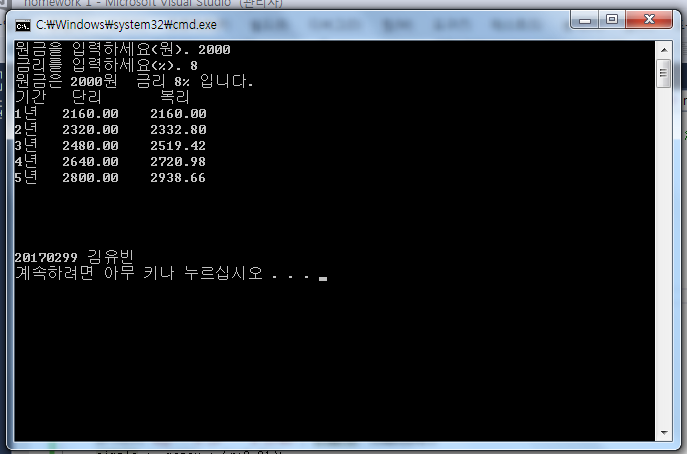
simple += money \* (r\*0.01);

compound \*= 1 + r \* 0.01;

printf("5년 %.2f %.2f\n\n\n\n\n", simple, compound);

printf("20170299 김유빈\n");

}



**<본 과제 – 반올림, 올림, 내림>**

해결방안: 주어진 실수x는 서식문자 %f를 통해서 소수점 6자리까지 출력된다. 실수x를 소수점 3자리까지 출력하기 위해서는 서식문자 %.3f를 사용해야 한다. %.3f는 소수점 4자리에서 반올림하여 주어진 값의 소수점 3자리까지 출력하는 속성을 지닌다. 이러한 속성을 이용하여 실수x의 반올림 값은 쉽게 구할 수 있다.

문제는 x의 내림과 올림 값을 구하는 것이다. 하지만 내림과 올림의 수학적 정의를 떠올리면 이 또한 쉽게 구할 수 있다. X에 대한 내림 값은 x – 0.0005를 소수점 4자리에서 반올림한 값이다. 예를 들어 123.4568에서 0.0005를 뺀 123.4563의 소수점 4자리에서의 반올림 값은 123.456이다. 또 다른 예로 123.4564에서 0.0005를 뺀 123.4559의 소수점 4자리에서의 반올림 값도 123.456이다. 이와 같이 x의 소수점 4자리 숫자가 5이상이면 반올림에 의해 버려지고, 5미만이면 반올림에 의해 올려진다. 이와 같은 방식으로 생각하면 x에 대한 올림은 x + 0.0005를 소수점 4자리에서 반올림한 값이다. 실수x의 소수점 4자리 숫자가 5이상이면 소수점 3자리 숫자를 1 증가시키고 반올림에 의해 버려진다. 이에 반해 x의 소수점 4자리 숫자가 5미만이면 반올림에 의해 그 값은 올려진다.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main()

{

float x;

printf("숫자를 입력하세요. ");

scanf("%f", &x);

printf("x = %f\n\n", x);

printf("x(반올림) = %.3f\n", x);

double floor = x - 0.0005;

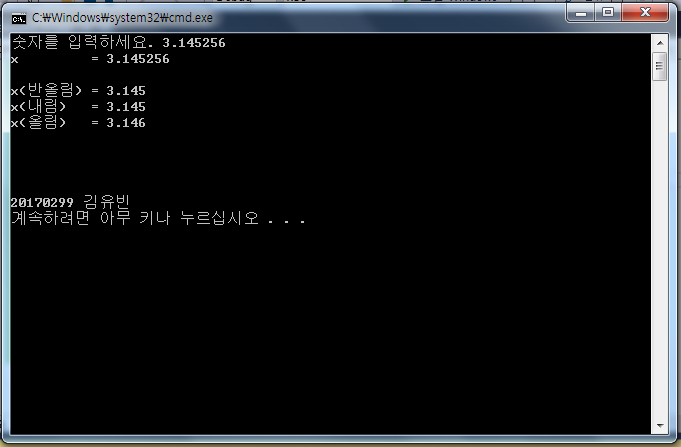
double ceil = x + 0.0005;

printf("x(내림) = %.3f\n", floor);

printf("x(올림) = %.3f\n\n\n\n\n", ceil);

printf("20170299 김유빈\n");

}



**<자유과제 (1)>**

출처 : 윤성우 열혈 C 프로그래밍 – 저자 윤성우

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int main(void)

{

int xpos1, ypos1; // 좌 상단의 x좌표의 변수를 xpos1, y좌표의 변수를 ypos1로 정의

int xpos2, ypos2; // 우 하단의 x좌표의 변수를 xpos2, y좌표의 변수를 ypos2로 정의

int area; /\* 두 점이 이루는 직사각형의 넓이에 해당하는 변수.

x, y좌표의 자료형을 정수형int로 하였기 때문에 직사각형 넓이 계산도 정수 ->area 자료형을 int형으로 정함\*/

printf("좌 상단의 x, y 좌표: ");

scanf("%d %d", &xpos1, &ypos1); /\*서식문자%d는 int형 정수를 입력할 때 쓰임.

x, y좌표에 해당하는 변수의 자료형이 int형이므로 서식문자%d를 사용\*/

printf("우 하단의 x, y 좌표: ");

scanf("%d %d", &xpos2, &ypos2);

area = (xpos2 - xpos1)\*(ypos2 - ypos1); // 오른쪽 계산의 결과값이 int형이므로

자료형이 int형인 area변수에 집어넣어짐

printf("두 점이 이루는 직사각형의 넓이는 %d입니다. \n", area);

//area의 값이 정수이므로 서식문자%d를 사용하여 int형 정수값을 출력

return 0;

}