아주대학교 프로그래밍언어

201821048 미디어학과 이서영

2019/10/06 if문, 연산자, 형변환

# 소스코드 및 주석

/\*

사용자로부터 졸업생 수와 취업자 수를 입력받아

취업률(%)을 계산하는 프로그램을 작성하시오.

아래 출력결과와 같이 소수점 둘째자리까지 출력하시오

취업률(%) = 취업자 수 / 졸업생 수 \* 100

순서 : 졸업자, 취업자 수 입력받기 -> 취업률 계산공식에 넣어서 취업률 구하기 -> 출력하기

요소 : 졸업자 수, 취업자 수, 취업률(실수)

포인트 : int형으로 받은 졸업자 수와 취업자 수를 계산해

double형인 취업률을 출력해주기

이 과정에서 형 변환이 필요하다. 형 변환의 종류는 5가지가 있다.

1. 확장변환 : 데이터를 입력할 때 double 형에 5를 넣어 5.0이 된다.

2. 축소변환 : 데이터를 입력할 때 int 형에 4.7을 넣어 4가 된다.

3. 배정문변환 : 정수 연산 시 char형, short형은 int형으로 변환한다.

4. 승급변환 : int형 5와 double형 1.5를 더하면 double형 6.5가 된다.

5. 강제형변환 : (int)6.7을 하면 강제로 형이 바뀌어 6이 된다.

\*/

#include <stdio.h> //std input, output을 포함한다.

int main() { //main 함수를 넣어 프로그램을 실행한다.

int graduate\_stu, employed\_stu = 0; //int 형 졸업자, 취업자 선언

printf("졸업생 수를 입력하시오 : ");

scanf\_s("%d", &graduate\_stu); //졸업자 입력받음 사람 숫자라서 int 형

printf("취업자 수를 입력하시오 : ");

scanf\_s("%d", &employed\_stu); //취업자 입력받음 사람 숫자라서 int 형

if (employed\_stu > graduate\_stu) { //취업자를 졸업자수보다 많이 입력했을 경우

printf("취업생 숫자가 졸업생 숫자보다 많을 수 없습니다"); //오류메시지 출력

return 0; //프로그램 종료

}

double employed\_rate; //double 형 취업률 선언 퍼센트로 둘째자리까지 출력 필요해서 double형

employed\_rate = (double)employed\_stu / (double)graduate\_stu \* 100; //계산식에 넣는다

//5강제형변환으로 졸업생수, 취업자수를 double로 바꾸어 정확한 값을 구하게 한다.

printf("취업률은 : %.2lf (%%) 입니다.", employed\_rate); //취업률 출력, 소수2번째 자리까지 하기

//%를 출력하기 위해서 %%를 쓴다.

return 0; //끝내기

}

/\*

3개의 정수값을 입력 받아서 3개의 정수값 중에서 최대값을 출력하는

프로그램을 작성하시오

조건연산자를 사용

순서 : 정수 3개를 입력받기 -> 큰 값을 결과에 저장하면서 비교하기 -> 결과값 출력하기

요소 : 첫번째 값, 두번쨰 값, 세번째 값, 결과 저장

포인트 : 입력받은 숫자 비교해서 큰 값 저장하기 > 연산자

연산자의 종류에는

대입 =

산술 + - \* / %

부호 + -

증감 ++ --

관계 > < == != >= <=

논리 && || !

조건 ?

콤마 ,

비트단위 & | ^ ~ << >>

sizeof sizeof

형변환 (type)

포인터 \* & []

구조체 . ->

가 있다.

그 중에서 관계연산자를 사용해 이 문제에서는 어떤 값이 큰지 비교하겠다.

그 후 조건연산라를 이용해 조건 안의 식이 맞으면 앞에 값을 아니면 뒤의 값을 저장한다.

이 문장은 if문을 이용해 조건을 나누어 할수도 있지만 조건연산자를 사용해 보고 싶었다.

\*/

#include <stdio.h> //std input, output을 포함한다.

int main() { //main 함수를 넣어 프로그램을 실행한다.

int first, second, third, result = 0; //int 형 첫번쨰, 두번쨰, 세번째 변수 선언

printf("3개의 정수를 입력하시오 : "); //입력 안내

scanf\_s("%d %d %d", &first, &second, &third); //세개의 값 입력받음 정수라서 int 형

result = (first > second) ? first : second; //조건 안의 식이 맞으면 앞에 값을 아니면 뒤의 값을 저장한다.

result = (result > third) ? result : third; //조건 안의 식이 맞으면 앞에 값을 아니면 뒤의 값을 저장한다.

printf("최대값 : %d", result); //최댓값 결과 출력

return 0;

}

/\*

cm로 표현된 키를 입력하여 피트와 인치로 변환하는 프로그램을 작성하시

오. 1피트는 12인치이고 1인치는 2.54cm 이다

순서 : 키를 입력 -> 피트와 인치 단위를 계산 -> 피트와 인치 출력

요소 : cm, feet, inch, 변환식

포인트 : 형변환하기, 변환식을 이용해 피트와 인치 구하기

1. 확장변환 : 데이터를 입력할 때 double 형에 5를 넣어 5.0이 된다.

2. 축소변환 : 데이터를 입력할 때 int 형에 4.7을 넣어 4가 된다.

3. 배정문변환 : 정수 연산 시 char형, short형은 int형으로 변환한다.

4. 승급변환 : int형 5와 double형 1.5를 더하면 double형 6.5가 된다.

5. 강제형변환 : (int)6.7을 하면 강제로 형이 바뀌어 6이 된다.

이 중에서 강제형변환을 사용한다.

피트 구하는 식 : CM\_PER\_FEET = (12 \* 2.54)을 이용해 cm / (int)CM\_PER\_FEET;식을 쓴다.

인치 구하는 식 : cm에서 피트를 뺸 값을 2.54로 나누어 남은 인치를 구한다.

\*/

#include <stdio.h> //std input, output을 포함한다.

int main() { //main 함수를 넣어 프로그램을 실행한다.

int cm; //cm인트형 선언

int feet; //feet인트형 선언

double inch; //inch 더블형 선언

const double CM\_PER\_FEET = (12 \* 2.54); //변환 비율 상수 변수로 선언

printf("키를 입력하시오(cm) : "); //키입력 안내

scanf\_s("%d", &cm); //cm 입력받기

feet = cm / (int)CM\_PER\_FEET; //CM\_PER\_FEET = (12 \* 2.54)을 이용해 cm / (int)CM\_PER\_FEET;식을 쓴다.

inch = (cm - feet \* CM\_PER\_FEET) / 2.54; // cm에서 피트를 뺸 값을 2.54로 나누어 남은 인치를 구한다.

printf("%dcm는 %d피트 %lf인치입니다.", cm, feet, inch); //cm(int), feet(int), inch(double)형을 출력한다.

return 0; //끝내기

}

/\*

cm로 표현된 키를 입력하여 피트와 인치로 변환하는 프로그램을 작성하시

오. 1피트는 12인치이고 1인치는 2.54cm 이다

순서 : 키를 입력 -> 피트와 인치 단위를 계산 -> 피트와 인치 출력

요소 : cm, feet, inch, 변환식

포인트 : 형변환하기, 변환식을 이용해 피트와 인치 구하기

1. 확장변환 : 데이터를 입력할 때 double 형에 5를 넣어 5.0이 된다.

2. 축소변환 : 데이터를 입력할 때 int 형에 4.7을 넣어 4가 된다.

3. 배정문변환 : 정수 연산 시 char형, short형은 int형으로 변환한다.

4. 승급변환 : int형 5와 double형 1.5를 더하면 double형 6.5가 된다.

5. 강제형변환 : (int)6.7을 하면 강제로 형이 바뀌어 6이 된다.

이 중에서 강제형변환을 사용한다.

피트 구하는 식 : CM\_PER\_FEET = (12 \* 2.54)을 이용해 (int)cm / CM\_PER\_FEET;식을 쓴다.

인치 구하는 식 : cm에서 피트를 뺸 값을 2.54로 나누어 남은 인치를 구한다.

\*/

#include <stdio.h> //std input, output을 포함한다.

int main() { //main 함수를 넣어 프로그램을 실행한다.

int cm; //cm인트형 선언

int feet; //feet인트형 선언

double inch; //inch 더블형 선언

const double CM\_PER\_FEET = (12 \* 2.54); //변환 비율 상수 변수로 선언

printf("키를 입력하시오(cm) : "); //키입력 안내

scanf\_s("%d", &cm); //cm 입력받기

if (cm <= 0) { //키 0보다 작거나 같은 값 입력 시

printf("키는 양수여야 합니다."); //오류메시지 출력하고 종료함.

return 0;

}

feet = (int)cm / CM\_PER\_FEET; //CM\_PER\_FEET = (12 \* 2.54)을 이용해 (int)cm / CM\_PER\_FEET;식을 쓴다.

inch = (cm - feet \* CM\_PER\_FEET) / 2.54; // cm에서 피트를 뺸 값을 2.54로 나누어 남은 인치를 구한다.

printf("%dcm는 %d피트 %lf인치입니다.", cm, feet, inch); //cm(int), feet(int), inch(double)형을 출력한다.

return 0; //끝내기

}

/\*

[과제#2]에서의 자판기 프로그램을 좀 더 개선하여

메뉴를 아래와 같이 만들고 각 메뉴 선택에 따라서

잔돈 계산을 수행하는 프로그램

이 때, 투입한 금액이 사용한 금액보다 작으면

“투입한 금액이 부족합니다.” 에러 처리

금액을 투입하세요:

아래 메뉴 중 하나를 선택하세요 :

1. 일반커피(250원)

2. 고급커피(500원)

3. 건강음료(1000원)

4. 에너지드링크(1500원)

선택한 메뉴를 몇 개나 구매하시겠습니까?

거스름돈은 아래와 같습니다

조건문(if문) 을 이용해야 구현 가능합니다

순서 : 돈 입력 -> 음료수 종류 선택 -> 맞는 선택, 돈 초과 안하면 돈 사용하기 -> 거스름돈 출력

요소 : kind\_of\_drinks, proper\_value, input\_money, cup\_of\_drinks, used\_money, total\_change, coin\_500, coin\_100, coin\_50

포인트 : 적절한지 if문을 이용해 판단하기 1. 음료수 종류, 2. 입력돈 사용돈이 초과 안하다도록 판단

\*/

#include <stdio.h> //std input, output을 포함한다.

int main() {

int input\_money, cup\_of\_drinks, used\_money, total\_change, coin\_500, coin\_100, coin\_50 = 0;

//입력할 돈, 잔 수, 사용된 돈, 거스름돈, 500원 개수, 100원 개수, 50원 개수

int kind\_of\_drinks, proper\_value = 0; //음료수 종류 선택, 적절한지 판단

printf("금액을 투입하세요 : "); //금액 투입 문구 출력

scanf\_s("%d", &input\_money); //사용자로부터 자판기에 넣는 돈 입력

printf("아래 매뉴 중 하나를 선택하세요\n"); //메뉴선택 출력

printf("1. 일반커피(250원)\n"); //1

printf("2. 고급커피(500원)\n"); //2

printf("3. 건강음료(1000원)\n"); //3

printf("4. 에너지드링크(1500원)\n"); //4

scanf\_s("%d", &kind\_of\_drinks);

printf("몇 개 뽑을까요? : "); ////잔 구매 수 문구 출력

scanf\_s("%d", &cup\_of\_drinks); //잔 구매 수

if (kind\_of\_drinks == 1) { //1번선택 시

used\_money = cup\_of\_drinks \* 250; //사용된 돈 = 잔수 곱하기 금액

proper\_value = ((input\_money - used\_money) >= 0) ? 1 : 0;

} //적절한지 판단, 넣은돈 - 사용한 돈이 양수, 같으면 적절, 음수면 부적절

if (kind\_of\_drinks == 2) { //2번선택 시

used\_money = cup\_of\_drinks \* 500; //사용된 돈 = 잔수 곱하기 금액

proper\_value = ((input\_money - used\_money) >= 0) ? 1 : 0;

}//적절한지 판단, 넣은돈 - 사용한 돈이 양수, 같으면 적절, 음수면 부적절

if (kind\_of\_drinks == 3) { //3번선택 시

used\_money = cup\_of\_drinks \* 1000; //사용된 돈 = 잔수 곱하기 금액

proper\_value = ((input\_money - used\_money) >= 0) ? 1 : 0;

}//적절한지 판단, 넣은돈 - 사용한 돈이 양수, 같으면 적절, 음수면 부적절

if (kind\_of\_drinks == 4) { //4번선택 시

used\_money = cup\_of\_drinks \* 1500; //사용된 돈 = 잔수 곱하기 금액

proper\_value = ((input\_money - used\_money) >= 0) ? 1 : 0;

}//적절한지 판단, 넣은돈 - 사용한 돈이 양수, 같으면 적절, 음수면 부적절

if (kind\_of\_drinks != 1 && kind\_of\_drinks != 2 && kind\_of\_drinks != 3 && kind\_of\_drinks != 4

|| proper\_value == 0) {

//음료 선택 시 1도 아니고 2도 아니고 3도 아니고 4도 아니면 다시 입력 프로그램 종료

//1 2 3 4 선택 후 넣은돈 - 사용한 돈이 음수면 다시 입력 프로그램 종료

printf("다시 입력하세요.");

return 0;

}

printf("총 사용금액은 %d 입니다.\n\n", used\_money); //적절하게 사용했다면 프로그램을 종료하지 않고 사용금액 출력

printf("\*\*\*\*\*\*\*거스름돈\*\*\*\*\*\*\*\*\*\*\n\n"); //거스름돈 안내 문구 출력

total\_change = input\_money - used\_money; //거스름돈 = 입력한 돈 - 사용금액

printf("총 거스름돈 %d 원\n", total\_change); //총 거스름돈 출력

coin\_500 = total\_change / 500; //500원 개수 구하기, 빼고 난 나머지에서 500원 최대지급

total\_change = total\_change - coin\_500 \* 500; //지급한 500원을 뺴고 나머지 저장

printf("500원짜리 동전 %d 개\n", coin\_500); //500원 개수 출력

coin\_100 = total\_change / 100; //100원 개수 구하기, 빼고 난 나머지에서 100원 최대지급

total\_change = total\_change - coin\_100 \* 100; //지급한 100원을 뺴고 나머지 저장

printf("100원짜리 동전 %d 개\n", coin\_100); //100원 개수 출력

coin\_50 = total\_change / 50; //50원 개수 구하기, 빼고 난 나머지에서 50원 최대지급

total\_change = total\_change - coin\_50 \* 50; //지급한 50원을 뺴고 나머지 저장

printf("50원짜리 동전 %d 개\n", coin\_50); //50원 개수 출력

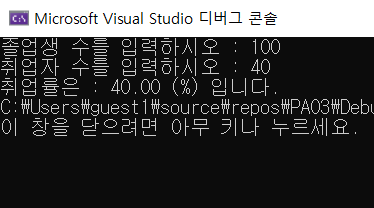
return 0;

}

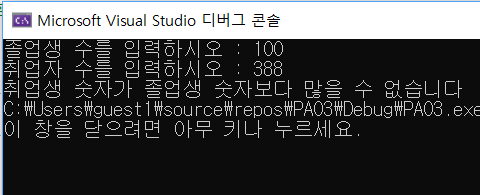
# 실행파일

실습 3

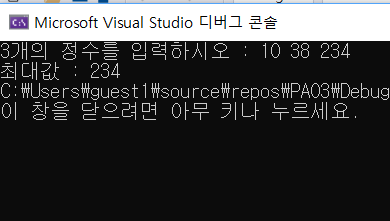
정상적으로 작동될 경우



졸업생보다 취업자 수가 많을 경우

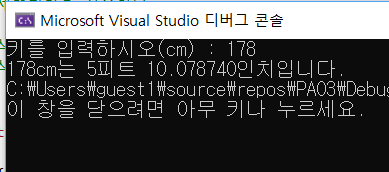


실습 4

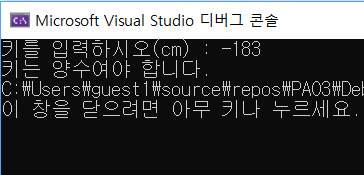


실습5

정상적으로 작동할 경우

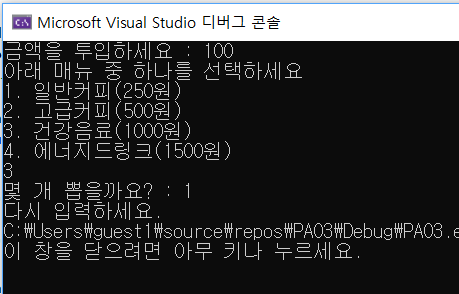


키에 0보다 작거나 같은 값을 넣을 경우

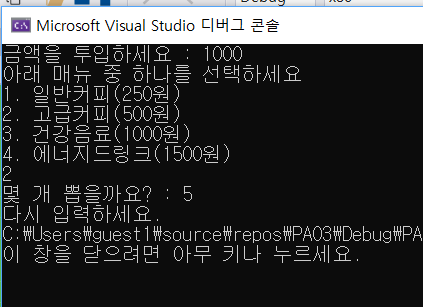


과제1

1. 금액이 적을 경우



1. 입력한 돈보다 사용금액이 클 경우



1. 정상적으로 돈을 넣어 작동될 경우

