**프로그래밍 언어 레포트3**

사회학과

201721899 송지민

**[과제#1] 태양빛 도달시간 수정 프로그램 작성**

1. **소스코드 및 주석**

// 태양빛 도달시간 수정 프로그램

#include <stdio.h> // 표준 입출력 함수 헤더 파일 선언

#define SEC\_PER\_MINUTE 60 // 1분은 60초, #define문을 이용하여 기호상수를 정의

/\*

#define 지시자는 특정 기호 상수를 정의한 치환문자열로 대체하는 역할을 하며, #define으로 시작되는 전처리 문장을 ‘매크로’라고 한다. 매크로는 인수를 가지지 않는 매크로 상수와 함수처럼 인수를 가지는 매크로 함수로 나뉜다.

기호상수로 정의하지 않아도 값은 동일하게 도출되지만, 기호상수로 정의해 두면 더 효율적이며 다른 이름과의 구별을 위해 대문자로 작성하는 것이 좋다. 기호상수를 만들기 위해서는 main() 함수 밖에 #define문을 이용하는 방법도 있으나 main()함수 안에 const를 이용하여 기호상수를 만들 수도 있다. \*/

int main(void) // 함수의 선언부(머리부)

{

double light\_speed = 300000; // 빛의 속도를 저장하는 변수, 선언과 동시에 초기값 입력

double distance = 149600000; // 태양과 지구 사이의 거리를 저장하는 변수, 선언과 동시에 초기값 입력

int time = 0; // 태양빛이 지구에 도달할 때까지 걸리는 시간, 선언과 동시에 초기화

int min = 0; // 태양빛이 지구에 도달할 때까지 걸리는 시간 중 '분'에 해당하는 값을 저장하는 변수, 선언과 동시에 초기화

int sec = 0; // 태양빛이 지구에 도달할 때까지 걸리는 시간 중 '초'에 해당하는 값을 저장하는 변수, 선언과 동시에 초기화

/\*

main() 함수 안에 선언하는 변수의 경우(지역 변수), 변수를 초기화하지 않으면 쓰레기 값이 들어있으므로 반드시 초기화하여 사용한다.

세 변수는 정수형으로 선언한다. 나머지 연산에 쓰이는 변수이기 때문이다.

여기서 time을 실수형으로 선언하면 나머지 연산에서 다음과 같은 오류가 발생한다.

[Error] invalid operands of types 'double' and 'int' to binary 'operator%'\*/

time = distance/light\_speed; // 태양빛이 지구까지 도달하는 데 걸리는 시간을 계산하는 식, 거리를 빛의 속도로 나눈다.

/\*

앞서 time을 정수로 선언했기 때문에 distance/light\_speed의 값은 정수로 내림 변환된다.

위에서 time을 실수로 선언하고 여기서 time = (int)(distance/light\_speed); 으로 변환해도 뒤의 나머지 연산에서 오류가 발생한다.

형변환 연산자는 변수 자체를 변화시키는 것이 아니라 잠시 그 형을 바꾸어 수식에서 사용하는 것뿐이므로

사용에 있어서 주의해야 한다.\*/

min = time/SEC\_PER\_MINUTE; // 태양빛이 지구에 도달하는 시간을 분으로 나타내기 위해 기호상수를 사용하여 60으로 나누기

sec = time%SEC\_PER\_MINUTE; // 태양빛이 지구에 도달하는 시간 중 초를 나타내기 위해 기호상수를 사용하여 나머지 연산하기

/\*

나머지 연산자는 중요한 연산자 중 하나이며, 이를 이용하여 홀수와 짝수를 쉽게 구분할 수 있다.

나머지 연산을 위해서는 연산에 사용되는 자료형이 int 형이어야 하는 것으로 보인다.

double 형을 사용할 경우 다음과 같은 오류가 발생하기 때문이다.

[Error] invalid operands of types 'double' and 'double' to binary 'operator%' \*/

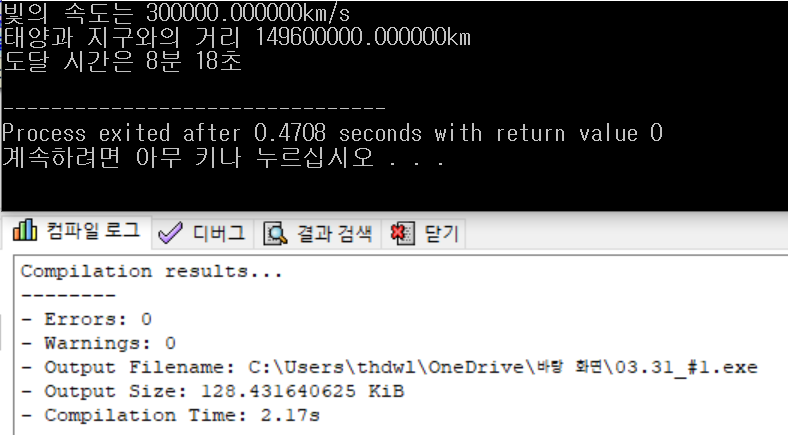
printf("빛의 속도는 %lfkm/s \n", light\_speed); // 빛의 속도를 출력, 실수형 변수이므로 변환명세는 %lf 혹은 %f 사용

printf("태양과 지구와의 거리 %lfkm \n", distance); // 태양과 지구 사이 거리를 출력, 실수형 변수이므로 변환명세는 %lf나 %f 사용

printf("도달 시간은 %d분 %d초\n", min, sec); // 도달시간을 n분 n초 형태로 출력. min, sec 모두 정수형 변수이므로 변환명세로 %d를 사용

return 0; // 0을 반환하여 main함수를 끝냄

}

1. **실행결과 및 컴파일 화면**
2. **부록**

* 소수점 두번째 자리까지 나타내 보기

위의 소스코드 중

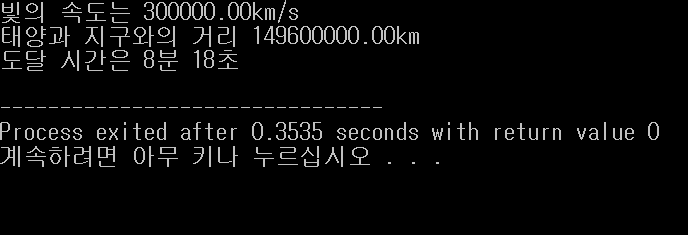
printf("빛의 속도는 %lfkm/s \n", light\_speed);

printf("태양과 지구와의 거리 %lfkm \n", distance);

이 두개의 출력 행을 다음과 같이 수정하면

printf("빛의 속도는 *%.2lf*km/s \n", light\_speed);

printf("태양과 지구와의 거리 %.*2lfkm* \n", distance);



실행화면에서 소수점 둘째 자리까지 나타나는 것을 확인하였다.

**[과제#2] 급여계산 프로그램 작성**

1. **소스코드 및 주석**

// 급여계산 프로그램

#include <stdio.h> /\* #inlcude 지시자, 특정 파일을 현 소스 파일에 포함하기 위한 지시자로

C 컴파일러가 제공하는 표준 라이브러리의 헤더 파일에 대해서는 <헤더파일명>을,

사용자가 직접만든 헤더 파일에 대해서는 "헤더 파일명"을 쓴다.\*/

int main(void) // 함수의 선언부

{

// 변수 선언 및 초기화하기. 쓰레기 값의 저장을 막기 위해 선언과 동시에 초기화

int bonbong=0;

int bonus=0;

int total=0; // 본봉, 보너스, 총수입액은 정수로 선언한다.

double tax\_rate=0;

double tax=0;

double real\_income=0; // 세율, 세금, 실수입액은 실수로 선언한다.

/\*

선언과 동시에 0으로 초기값을 입력했기 때문에 값을 입력 받기 전에 처리를 수행하면

공제액과 실수입액이 0.000000으로 출력되는 오류가 발생하므로 주의해야 한다.(부록1 캡처 화면 참고)

세율, 세금, 실수입액은 실수형 변수이기 때문에 0.0으로 올림 변환되었을 것이다.\*/

// 본봉, 보너스, 세율 입력받기

printf("\*\*\*\*\*\*\*\*\* 급여계산 프로그램<단위: 만원>\*\*\*\*\*\*\*\*\*\*\*\n"); // 프로그램 이름 출력하기

/\* 바로 아래에 printf() 함수, 즉 출력함수가 오므로 개행문자를 이용하여 줄바꿈을 해주어야 한다.

출력 -> 입력 순으로 행이 위치하는 경우 출력함수 내에 따로 개행 문자를 입력할 필요가 없는 것으로 보인다. 그러나 개행 문자를 입력해도 무방하다.(부록2 참고) 단, scanf()함수에 개행 문자를 입력할 경우 오류가 발생하므로 주의해야 한다.(부록3 참고)\*/

printf("본봉을 입력하세요:");

scanf("%d", &bonbong); // 본봉 입력 받기. 본봉은 정수로 선언하였으므로 %d 변환명세를 이용해야 함

printf("보너스를 입력하세요:");

scanf("%d", &bonus); // 보너스 입력받기. 보너스를 정수로 선언하였으므로 %d 변환명세를 이용해야 함

printf("공제할 세율<%%>을 입력하세요:"); /\* <%>를 출력하기 위해서 <%%>를 사용하여야 함을 주의해야 한다.

<%>라고 작성할 경우 출력 화면에는 <>가 나타나게 된다.\*/

scanf("%lf", &tax\_rate); /\* 세율 입력받기. 세율은 실수로 선언하였으므로 %lf 변환명세를 이용해야 함

입력 받은 변수를 저장하기 위해 주소 연산자 &를 사용해야 함을 주의하기\*/

// 자료 처리하기

total=bonbong+bonus; // 총수입액 계산하기

tax=total\*tax\_rate/100; /\* 세금 계산하기. 세율을 '9'와 같은 형태로 입력받기 때문에

퍼센트의 형태로 계산하기 위해서 tax\_rate/100을 반드시 포함해야 한다.

tax\_rate\*0.01형태로 작성하여도 값은 동일하다.\*/

real\_income=total-tax; // 총수입액에서 세금을 차감하여 실수입액 계산하기

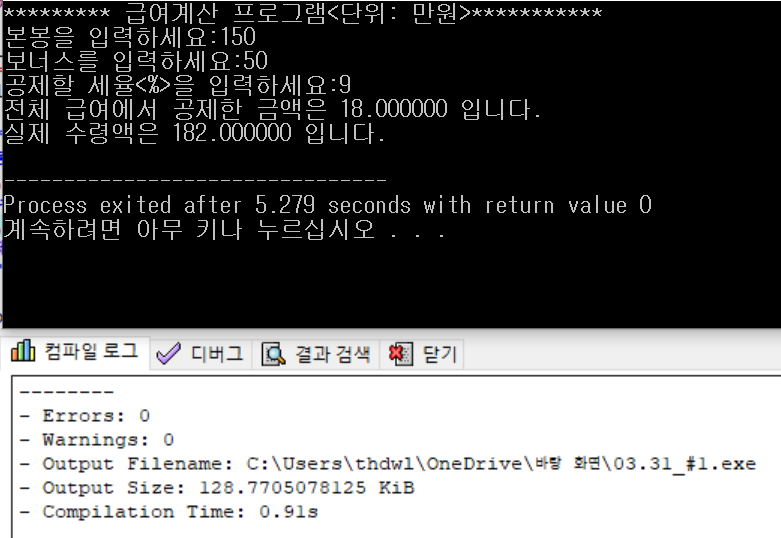
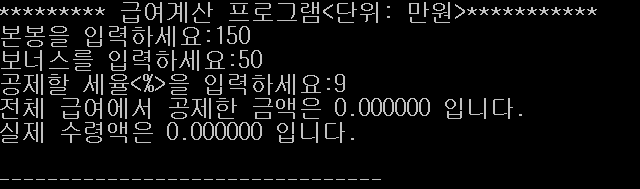
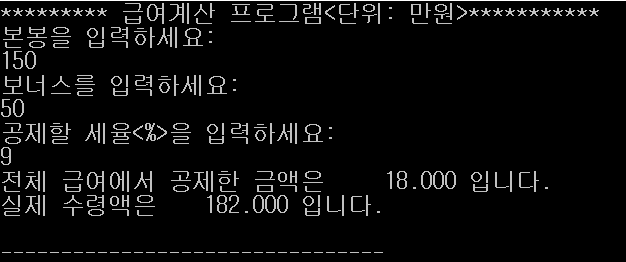
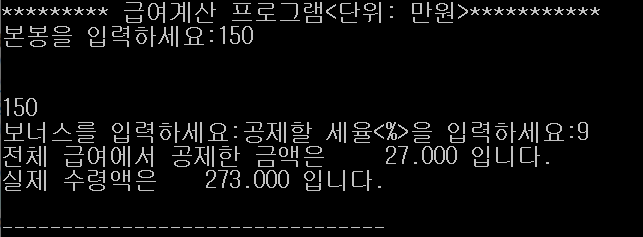
// 결과 출력하기

printf("전체 급여에서 공제한 금액은 %lf 입니다.\n", tax); // 세금 출력하기, tax는 실수형 변수이므로 변환명세는 %lf를 사용한다.

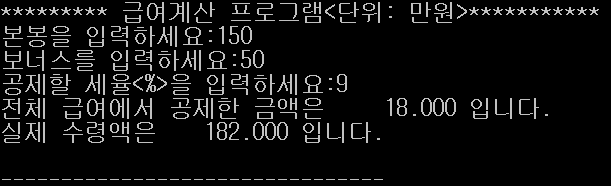
printf("실제 수령액은 %lf 입니다.\n", real\_income); // 실수입액 출력하기, real\_income은 실수형 변수이므로 변환명세는 %lf를 사용한다.

return 0; // 메인 함수에 0 반환하여 끝내기

}

1. **실행결과 및 컴파일 화면**
2. **부록**
3. 0으로 값을 초기화한 다음 바로 변수를 처리했을 때의 오류 화면
4. printf()에 개행 문자 입력하기
5. scanf()에 개행 문자 입력했을 때의 오류 화면

scanf(“%d\n”, bonus); 의 형태로 작성한 결과 위와 같은 오류가 발생하였다.

1. 출력 시 소수점 세 번째 자리까지 나타내고 총 10칸 지정해보기

과제#2의 결과 출력 소스코드를 다음과 같이 변경하면

printf("전체 급여에서 공제한 금액은 %10.3lf 입니다.\n", tax);

printf("실제 수령액은 %10.3lf 입니다.\n", real\_income);

실행 결과는 위의 화면과 같다.

**[과제#3] 평을 제곱미터로 환산하는 프로그램 작성**

1. **소스코드 및 주석**

// 평을 제곱미터로 환산하는 프로그램

#include <stdio.h> // 전처리문, 표준 입출력 함수 헤더 파일을 포함하라는 의미

/\*

전치리기란, 컴파일러가 소스 파일을 컴파일하기 전에 수행되는 프로그램으로,

프로그래머가 작성한 소스 파일이 컴파일될 수 있도록 준비하는 역할을 한다.

한 행에 지시자 하나만 쓸 수 있으며 컴파일러가 아닌 전처리기가 처리하므로

명령어 끝에 세미콜론(;)을 붙이지 않는다.\*/

int main(void) // 함수의 선언부

/\*

C프로그램에서 가장 먼저 실행되는 함수이며 가장 중요한 함수이다.

함수의 이름은 반드시 소문자 main()이어야 하며 main()함수는

C프로그램 당 하나만 작성하여아 한다. 여러 개를 만들면 링크 오류가 발생한다.\*/

{

// 변수 선언 및 초기화

int pyeong = 0;

double sqmeter = 0; // 쓰레기 값으로 인한 오류를 막기 위해 초기화

const double SQMETER\_PER\_PYEONG=3.3058; // 1평 = 3.3058 제곱미터이므로 이를 기호상수로 선언하기

/\*

#define 문장과 달리 메인 함수 내에서 변수 선언 앞에 const를 붙이면 상수가 된다.

선언 시 const가 붙여지면 초기화된 후에 그 값이 변경될 수 없다.

일반적으로 기호상수 선언 시에는 #define 보다 const를 사용하는 편이 장점이 많으므로

cosnt에 익숙해지는 것이 좋다.\*/

// 자료 입력 받기

printf("평을 입력하세요: "); // 출력함수 printf() 를 이용하여 지시사항 출력하기

scanf("%d", &pyeong); // 입력함수 scanf() 를 이용하여 평 입력 받기. 평은 정수형 자료이므로 변환명세는 %d 이다.

/\*

scanf()는 getchar()나 get\_s()와 달리 문자열을 숫자의 형태로 자동적으로 변환시켜 준다.

scanf()도 printf()와 마찬가지로 scanf("%2d%2d", &pyeong, &pyeong2)와 같이 필드폭을 지정하여 사용할 수 있고

이 경우 1234를 12와 34처럼 두 글자씩 나누어서 읽는다.\*/

// 자료 처리

sqmeter = pyeong\*SQMETER\_PER\_PYEONG; // 제곱미터를 기호상수와 입력 받은 평을 이용하여 계산하기

// 결과 출력

printf("%.2lf평방미터입니다", sqmeter); // 소수점 두번째 자리까지 포함하여 출력하기

/\*

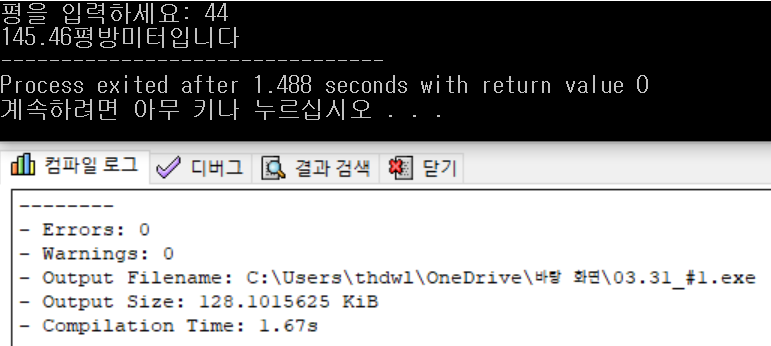
이 때 소수점에 대한 표현은 변환명세에서의 플래그값을 이용한다.

"%필드폭d"(출력될 전체 칸 규정), "%+필드폭d"(부호 출력), "%-필드폭d"(왼쪽 정렬)와 같은 형태로 사용한다.\*/

return 0; // 메인 함수에 0 반환하여 끝내기

}

1. **실행결과 및 컴파일 화면**



1. **부록**

* 변환명세 플래그값과 기호상수를 활용하여 4가지의 평을 입력 받아 제곱미터 출력하기
* 소스코드(메인 함수 내의 소스코드만 기술함)

{

// 변수 선언 및 초기화

int pyeong = 0;

int pyeong2=0;

int pyeong3=0;

int pyeong4=0;

double sqmeter = 0; // 쓰레기 값으로 인한 오류를 막기 위해 초기화

const double SQMETER\_PER\_PYEONG=3.3058; // 1평 = 3.3058 제곱미터이므로 이를 기호상수로 선언하기

double sqmeter2=0;

double sqmeter3=0;

double sqmeter4=0;

// 자료 입력 받기

printf("평을 입력하세요: ");

printf("평2를 입력하세요: ");

scanf("%d", &pyeong2);

printf("평3을 입력하세요: ");

scanf("%d", &pyeong3);

printf("평4를 입력하세요: ");

scanf("%d", &pyeong4);

// 자료 처리

sqmeter = pyeong\*SQMETER\_PER\_PYEONG; // 제곱미터를 기호상수와 입력 받은 평을 이용하여 계산하기

sqmeter2 = pyeong2\*SQMETER\_PER\_PYEONG;

sqmeter3 = pyeong3\*SQMETER\_PER\_PYEONG;

sqmeter4 = pyeong4\*SQMETER\_PER\_PYEONG;

// 결과 출력

printf("평1을 제곱미터로 환산하면:%10.2lf\n", sqmeter); // 출력 칸을 동일하게 하여 보기 좋게 출력하기

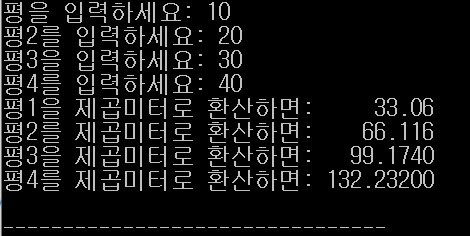
printf("평2를 제곱미터로 환산하면:%10.3lf\n", sqmeter2);

printf("평3을 제곱미터로 환산하면:%10.4lf\n", sqmeter3);

printf("평4를 제곱미터로 환산하면:%10.5lf\n", sqmeter4);

return 0; // 메인 함수에 0 반환하여 끝내기

}

* 실행화면