

10 주차 **중간고사 대체** Report

프로그램 문제 1) 다음 내용에 맞는 프로그램을 작성하시오.

- void 형 사용자 정의 함수: 인치를 센티미터로 환산하는 함수, 평을 평방미터로 환산하는 함수
- 1 인치 = 2.5399 센티미터, 1 평 = 3.3058 평방미터
- 메인 함수에서 문자형 변수 1 개와 실수형(float) 변수 1 개 선언
- do~while 문으로 프로그램 무한 반복
- if~else 문으로 대소 문자 구별 없이 A, B, Q 를 판별하여 분기하고 그 외 알파벳은 예외 처리
- A, B, Q 외의 알파벳 입력 시 continue 문 사용하여 다시 입력 요청
- 입력된 알파벳에 따라 분기문 수행: A 는 인치를 센티미터로 환산, B 는 평을 평방미터로 환산
- 프로그램 종료는 알파벳 대소 문자 구별 없이 Q 사용
- 알파벳은 메인 함수에서 입력받아 사용자 정의 함수에 전달
- 단위 환산 결과는 사용자 정의 함수에서 출력
- 기타 사항은 [출력 예시] 참조



프로그램 문제 1. 실행화면

➡ 출력 예시 - 1 : 키보드로 a 입력 ⇒ 25 입력

A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료

선택 : **a**

A. 인치를 센티미터로 환산 프로그램

문> 인치(inch)를 입력하세요 : **25**

[인치를 센티미터로 환산 함수 호출]

답> 25.00인치(inch)는 63.50센티미터(cm)

성공적으로 환산을 수행하였습니다.

A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료

선택 : **—**

➡ 출력 예시 - 2 : 키보드로 Q 입력

A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료

선택 : **Q**

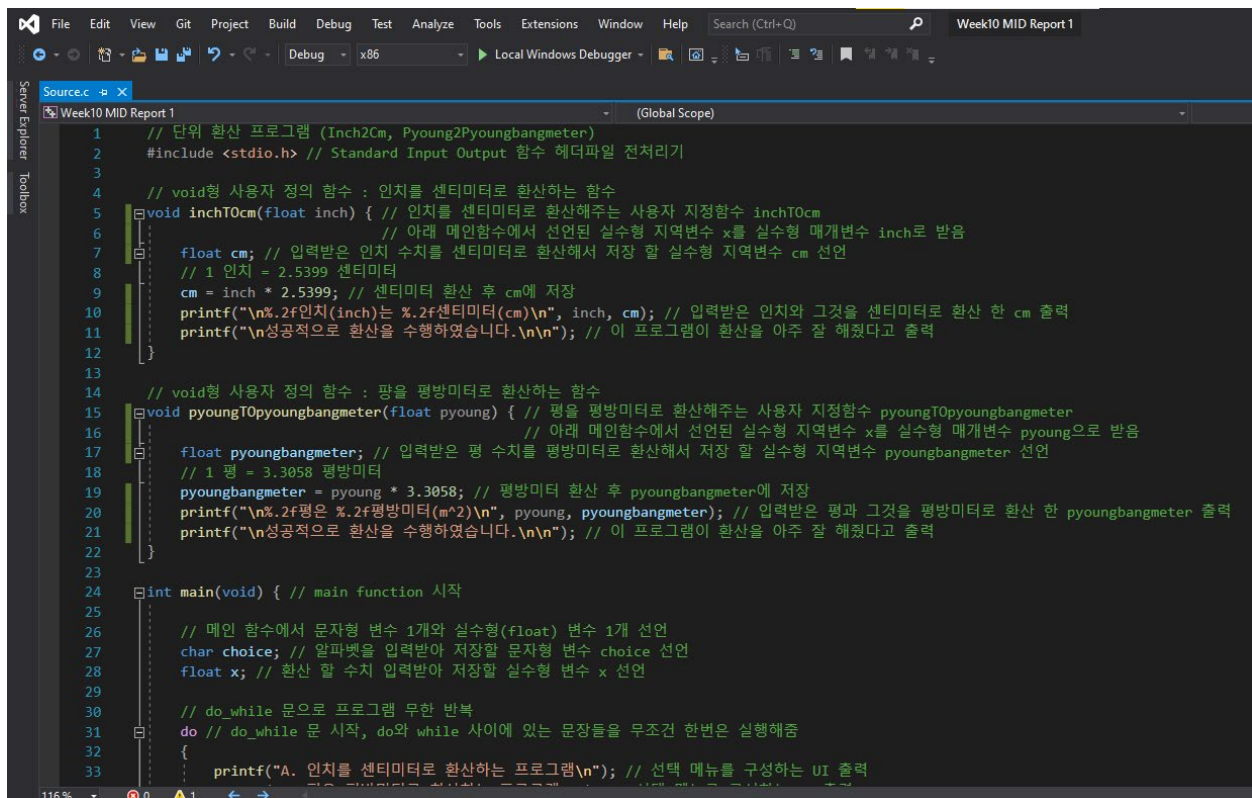
프로그램을 종료합니다.

1. 문제 분석

단위 환산을 해주는 프로그램이다. 문제 조건에 따라 void 형으로 인치를 센티미터로 환산해주거나, 평을 평방미터로 환산해줄도록 하는 사용자 정의 함수를 구성하여 그 내에서 환산 연산(1 인치 = 2.5399 센티미터, 1 평 = 3.3058 평방미터)이 이루어 지도록 하고 환산 결과를 출력해줄도록 하자. void 형 함수이기 때문에 리턴 값이 필요하지 않을 것이다. 그리고 문제조건에 따라 메인 함수 내에서 do_while 문을 통해 무한 반복을 해야 하므로 while 의 조건식을 1 로 설정하면 되겠다. 그리고 do 와 while 사이에 인치를 센티미터로 환산하는 기능과 평을 평방미터로 환산하는 기능을 알파벳으로 선별할 수 있는 if 문을 설정하여 사용자로부터 a 또는 A 를 입력 받게 되면 인치를 센티미터로 환산하는 함수를 실행할 수 있도록 해주고, b 또는 B 를 입력 받게 되면 평을 평방미터로 환산하는 함수를 실행할 수 있도록 해주자. 그리고 q 또는 Q 를 입력 받게 되면 break 을 이용해 do_while 문을 탈출해 프로그램을 종료하도록 해주고, a 또는 A, b 또는 B, q 또는 Q 이외의 알파벳을 입력 받게 되었을 때, 잘못된 입력이라 사용자에게 알려주면서 continue 를 이용해 다시 입력을 받게 해주자. 이런 알파벳을 입력 받아 저장할 문자형 변수를 선언해주고, 환산할 수치를 입력 받아 저장할 실수형 변수를 선언해주자.

2. 프로그램 소스 + 각 라인 주석 설명

(Line 1 ~ Line 33)



```
1 // 단위 환산 프로그램 (Inch2Cm, Pyoung2Pyoungbangmeter)
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3
4 // void형 사용자 정의 함수 : 인치를 센티미터로 환산하는 함수
5 void inchToCm(float inch) { // 인치를 센티미터로 환산해주는 사용자 지정함수 inchToCm
6     // 아래 메인함수에서 선언된 실수형 지역변수 x를 실수형 매개변수 inch로 받음
7     float cm; // 입력받은 인치 수치를 센티미터로 환산해서 저장 할 실수형 지역변수 cm 선언
8     // 1 인치 = 2.5399 센티미터
9     cm = inch * 2.5399; // 센티미터 환산 후 cm에 저장
10    printf("\n%.2f인치(inch)는 %.2f센티미터(cm)\n", inch, cm); // 입력받은 인치와 그것을 센티미터로 환산 한 cm 출력
11    printf("\n성공적으로 환산을 수행하였습니다.\n\n"); // 이 프로그램이 환산을 아주 잘 해줬다고 출력
12 }
13
14 // void형 사용자 정의 함수 : 평을 평방미터로 환산하는 함수
15 void pyoungToPyoungbangmeter(float pyoung) { // 평을 평방미터로 환산해주는 사용자 지정함수 pyoungToPyoungbangmeter
16     // 아래 메인함수에서 선언된 실수형 지역변수 x를 실수형 매개변수 pyoung으로 받음
17     float pyoungbangmeter; // 입력받은 평 수치를 평방미터로 환산해서 저장 할 실수형 지역변수 pyoungbangmeter 선언
18     // 1 평 = 3.3058 평방미터
19     pyoungbangmeter = pyoung * 3.3058; // 평방미터 환산 후 pyoungbangmeter에 저장
20    printf("\n%.2f평은 %.2f평방미터(m^2)\n", pyoung, pyoungbangmeter); // 입력받은 평과 그것을 평방미터로 환산 한 pyoungbangmeter 출력
21    printf("\n성공적으로 환산을 수행하였습니다.\n\n"); // 이 프로그램이 환산을 아주 잘 해줬다고 출력
22 }
23
24 int main(void) { // main function 시작
25
26     // 메인 함수에서 문자형 변수 1개와 실수형(float) 변수 1개 선언
27     char choice; // 알파벳을 입력받아 저장할 문자형 변수 choice 선언
28     float x; // 환산 할 수치 입력받아 저장할 실수형 변수 x 선언
29
30     // do_while 문으로 프로그램 무한 반복
31     do // do_while 문 시작, do와 while 사이에 있는 문장들을 무조건 한번은 실행해줌
32     {
33         printf("A. 인치를 센티미터로 환산하는 프로그램\n"); // 선택 메뉴를 구성하는 UI 출력
```

(Line 30 ~ Line 61)

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Week10 MID Report 1
Debug x86 Local Windows Debugger
Source.c
Week10 MID Report 1 (Global Scope)
30 // do_while 문으로 프로그램 무한 반복
31 do // do_while 문 시작, do와 while 사이에 있는 문장들을 무조건 한번은 실행해줌
32 {
33     printf("A. 인치를 센티미터로 환산하는 프로그램\n"); // 선택 메뉴를 구성하는 UI 출력
34     printf("B. 평을 평방미터로 환산하는 프로그램\n"); // 선택 메뉴를 구성하는 UI 출력
35     printf("Q. 프로그램 종료\n"); // 선택 메뉴를 구성하는 UI 출력
36     printf("\n"); // 선택 메뉴를 구성하는 UI 출력
37     printf("선택 : "); // 선택 메뉴를 구성하는 UI 출력, 사용자에게 메뉴를 선택해라는 메시지 출력
38     scanf_s("%c", &choice); // 선택 메뉴 알파벳을 입력받아 문자형 변수 choice에 저장
39
40     // if_else 문으로 프로그램 대소 문자 구별 없이 A, B, Q를 판별하여 분기하고 그 외 알파벳은 예외 처리
41     // 입력된 알파벳에 따라 분기문 수행 : A는 인치를 센티미터로 환산
42     if (choice == 'a' || choice == 'A') { // 만약 입력한 알파벳이 a or A 이라면
43         printf("A. 인치를 센티미터로 환산하는 프로그램\n"); // 선택한 메뉴가 인치를 센티미터로 환산하는 메뉴라는 것을 출력해주고
44         printf("인치를 입력하세요 : "); // 사용자에게 센티미터로 환산할 인치의 수치를 입력해라는 메시지 출력
45         scanf_s("%f", &x); // 사용자로부터 입력 받은 수치를 실수형 변수 x에 저장
46         printf("\n인치를 센티미터로 환산 함수 호출\n"); // 사용자 지정 함수를 호출 할 것이라는 메시지 출력
47         // 알파벳은 메인 함수에서 입력받아 사용자 정의 함수에 전달
48         // 단위 환산 결과는 사용자 정의 함수에서 출력
49         inchToCm(x); // 인치를 센티미터로 환산해주는 사용자 지정함수 inchToCm의 실수형 매개변수 inch 에 x값을 올려보냄
50     }
51
52     // 입력된 알파벳에 따라 분기문 수행 : B는 평을 평방미터로 환산
53     else if (choice == 'b' || choice == 'B') { // 만약 입력한 알파벳이 b or B 이라면
54         printf("B. 평을 평방미터(m^2)로 환산하는 프로그램\n"); // 선택한 메뉴가 평을 평방미터로 환산하는 메뉴라는 것을 출력해주고
55         printf("평을 입력하세요 : "); // 사용자에게 평방미터로 환산할 인치의 수치를 입력해라는 메시지 출력
56         scanf_s("%f", &x); // 사용자로부터 입력 받은 수치를 실수형 변수 x에 저장
57         printf("\n평을 평방미터로 환산 함수 호출\n"); // 사용자 지정 함수를 호출 할 것이라는 메시지 출력
58         // 알파벳은 메인 함수에서 입력받아 사용자 정의 함수에 전달
59         // 단위 환산 결과는 사용자 정의 함수에서 출력
60         pyoungTopyoungbangmeter(x); // 평을 평방미터로 환산해주는 사용자 지정함수 pyoungTopyoungbangmeter의 실수형 매개변수 pyoung 에 x값을 올려보냄
61     }
```

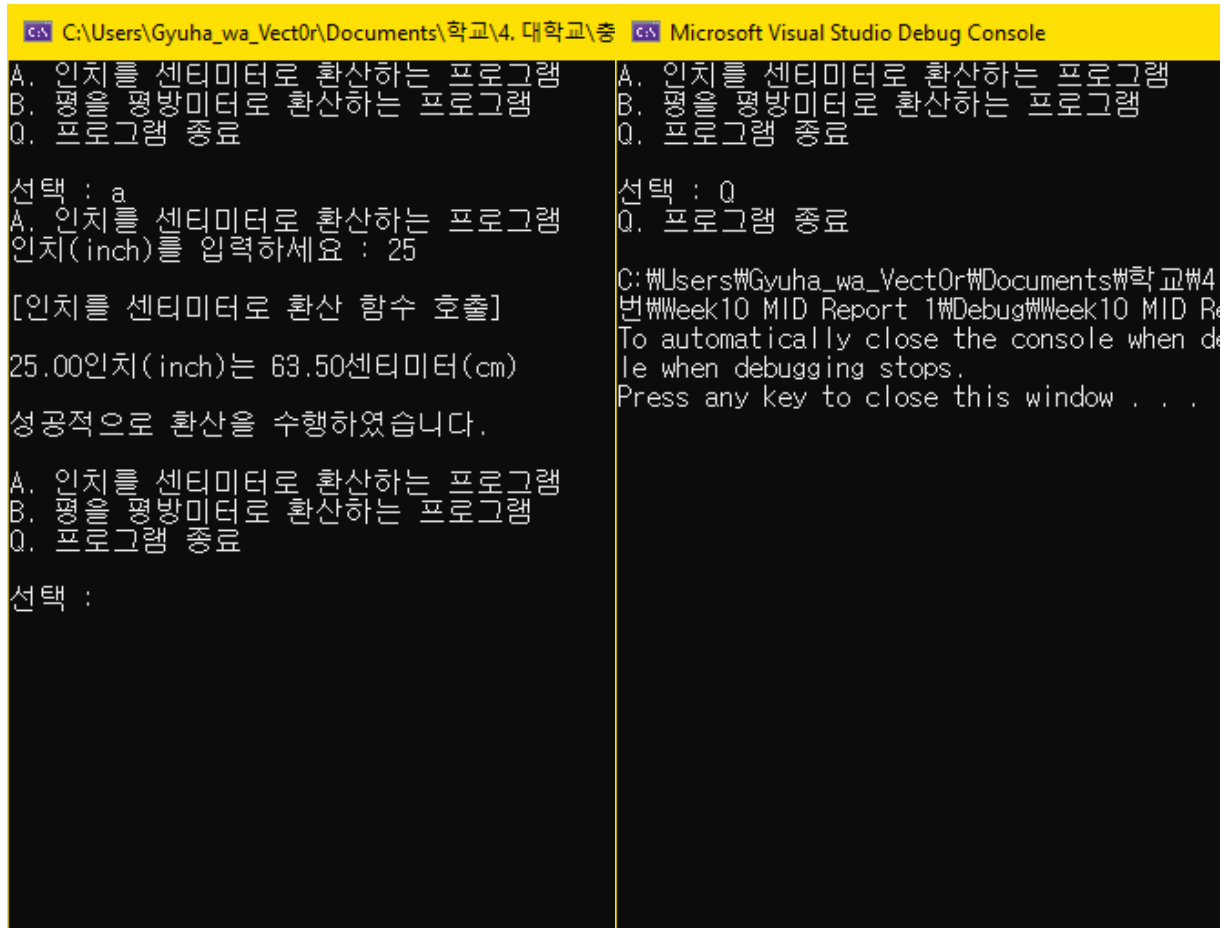
(Line 51 ~ Line 78)

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Week10 MID Report 1
Debug x86 Local Windows Debugger
Source.c
Week10 MID Report 1 (Global Scope)
51 // 입력된 알파벳에 따라 분기문 수행 : B는 평을 평방미터로 환산
52 else if (choice == 'b' || choice == 'B') { // 만약 입력한 알파벳이 b or B 이라면
53     printf("B. 평을 평방미터(m^2)로 환산하는 프로그램\n"); // 선택한 메뉴가 평을 평방미터로 환산하는 메뉴라는 것을 출력해주고
54     printf("평을 입력하세요 : "); // 사용자에게 평방미터로 환산할 인치의 수치를 입력해라는 메시지 출력
55     scanf_s("%f", &x); // 사용자로부터 입력 받은 수치를 실수형 변수 x에 저장
56     printf("\n평을 평방미터로 환산 함수 호출\n"); // 사용자 지정 함수를 호출 할 것이라는 메시지 출력
57     // 알파벳은 메인 함수에서 입력받아 사용자 정의 함수에 전달
58     // 단위 환산 결과는 사용자 정의 함수에서 출력
59     pyoungTopyoungbangmeter(x); // 평을 평방미터로 환산해주는 사용자 지정함수 pyoungTopyoungbangmeter의 실수형 매개변수 pyoung 에 x값을 올려보냄
60 }
61
62
63 // 프로그램 종료는 알파벳 대소 문자 구별 없이 Q 사용
64 else if (choice == 'q' || choice == 'Q') { // 만약 입력한 알파벳이 q or Q 이라면
65     printf("Q. 프로그램 종료\n"); // 선택한 메뉴가 프로그램을 종료한다는 메뉴라는 것을 출력해주고
66     break; // do_while 문을 곧바로 탈출하여 아래에 0으로 리턴 후 프로그램 종료
67 }
68
69 // A, B, Q 외의 알파벳 입력 시 continue 문 사용하여 다시 입력 요청
70 else { // 만약 입력한 알파벳이 a, A, b, B, q, Q 가 아닌 다른 알파벳 이라면
71     printf("\n올바른 명령을 입력하십시오.\n\n"); // 사용자에게 제대로 입력해라는 메시지 출력 후
72     continue; // do_while 문의 처음으로 되 돌아감
73 }
74
75 } while (1); // while의 조건이 1이므로 이는 do와 while 안에 있는 내용을 무한 반복하게 된다.
76
77 return 0; // main function 이 0으로 반환
78 }
```

3. 실행화면 캡처

(출력 예시 1: 키보드로 a 입력 -> 25 입력)

(출력 예시 2: 키보드로 Q 입력)



```
C:\Users\Gyuha_wa_Vect0r\Documents\학교\4. 대학교\중 C:\Users\Gyuha_wa_Vect0r\Documents\학교\4. 대학교\중 Microsoft Visual Studio Debug Console
A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료
선택 : a
A. 인치를 센티미터로 환산하는 프로그램
인치(inch)를 입력하세요 : 25
[인치를 센티미터로 환산 함수 호출]
25.00인치(inch)는 63.50센티미터(cm)
성공적으로 환산을 수행하였습니다.
A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료
선택 :
A. 인치를 센티미터로 환산하는 프로그램
B. 평을 평방미터로 환산하는 프로그램
Q. 프로그램 종료
선택 : Q
Q. 프로그램 종료
C:\Users\Gyuha_wa_Vect0r\Documents\학교\4. 대학교\중 Week10 MID Report 1\Debug\Week10 MID Report 1
To automatically close the console when debugging stops, please click OK here.
Press any key to close this window . . .
```

4. 프로그램 동작 설명

출력 예시에 맞춰 동작 설명을 해 보겠다.

(출력 예시 1)

메인 함수에서 메뉴 선택 알파벳 a 를 입력받아 문자형 변수 choice 에 저장한다.

choice == 'a' 이므로 if 문의 조건에 따라 인치를 센티미터로 환산하는 쪽으로 진행되어야 한다. 사용자로 부터 센티미터로 환산할 인치를 입력 받고, 위에서 마련한 사용자 지정 함수 inchToCm 의 매개변수로 그 입력 받은 것을 넘겨준다. 사용자 정의 함수 inchToCm 에서 입력 받은 인치 값에다 2.5399 를 곱해주고 inchToCm 내의 실수형 지역변수 cm 에 저장해준다.

그리고 입력받은 인치 값과 환산한 센티미터 값을 출력해준다. while 의 조건문이 1 이므로 환산을 선택하는 메뉴를 제시해주면서 그곳에서 다시 시작한다.

(출력 예시 2)

메인 함수에서 메뉴 선택 알파벳 Q 를입력받아 문자형 변수 choice 에 저장한다.

choice == 'Q' 이므로 if 문의 조건에 따라 프로그램을 종료하는 쪽으로 진행되어야 한다.

친절하게 프로그램을 종료한다는 메시지를 출력해주고 break 문에 의해 do_while 문에서 빠져나가 메인 함수에 0 으로 반환되면서 프로그램이 종료가 된다.

프로그램 문제 2) 다음 내용에 맞는 프로그램을 작성하시오.

- 매개변수만 구성 요소로 가지는 사용자 정의 함수 1 개 정의
- 메인 함수에서는 정수형 변수 3 개 선언
- 가계부 입력 횟수 (1 ~ 10)는 if ~ else 문으로 허용 범위 판별
- 입력 횟수 허용 범위가 아닐 경우 goto 문 사용하여 다시 입력
- for 문으로 입력 횟수에 따라 반복 수행
- 사용자 정의 함수에서 정적 지역 변수 1 개 선언하여 입출금 내역 연산하여 출력
- 기타사항은 [출력 예시] 참조



프로그램 문제 2. - 실행화면

➡ 출력 예시 : 13 입력 ⇒ -5 입력 ⇒ 2 입력 ⇒ 500000 입력 ⇒ -25000 입력

■ 정적 변수와 사용자 정의 함수를 사용한 가계부 프로그램

입출금 횟수 지정 (1~10) : 13

허용 범위가 아닙니다.
다시 입력하세요.

입출금 횟수 지정 (1~10) : -5

허용 범위가 아닙니다.
다시 입력하세요.

입출금 횟수 지정 (1~10) : 2

< 1 회 실행 >

현금 : 500000

입금

500000

출금

잔액

500000

< 2 회 실행 >

현금 : -25000

입금

출금

25000

잔액

475000

총 2회를 실행 후 프로그램 종료

1. 문제 분석

정적 변수와 사용자 정의 함수를 사용하여 입금과 출금한 비용을 잔액에 대해 연산하는 프로그램이다. 가계부 입력 횟수를 입력 받아 연산으로 넘어갈지 안 넘어갈지 결정하기 위해 `if_else` 문을 이용하고 그 횟수에 따라 반복문을 돌려주도록 하자. 그 반복문은 입출금 연산을 해주는 사용자 정의 함수를 만들어 그 내에서 돌리도록 해주자. 그 사용자 정의 함수는 매개변수만 구성 요소로 가져야한다. 만약 횟수가 1~10 사이에 들지 않으면 다시 입력해라는 메시지를 띄우고 다시 입력하도록 돌아가줘야 하는데, 이를 `goto` 문을 통해 횟수를 다시 입력 받도록 하자. 이러한 내용을 메인함수에서 구현하기 위해서는 3 개의 변수가 필요할 것이다. 입력 받은 입출금 횟수를 저장 할 정수형 변수, 입출금 연산을 해주는 함수 내에서 입력받은 입출금 횟수에 따른 반복문을 돌리기 위한 정수형 변수, 입출금 연산을 해주는 함수 내에서 입출금 할 금액을 입력받고 연산에 사용되는 정수형 변수가 선언 되어야 할 것이고 이 세가지는 입출금 연산을 해주는 함수의 매개변수와 연결이 되어야 할 것이다. 입출금 연산을 해주는 함수 내에서는 입출금 내역을 연산하여 잔액을 출력해주는 지역 변수가 있어야 하는데, 매 턴마다 초기화 되면 안되므로 정적지역변수로 선언해야 할 것이다. 앞서 반복문을 이 입출금 연산을 해주는 함수 내에 위치하도록 하고 입력받은 횟수만큼 1 부터 반복문을 돌리도록 한다. 그러나 실행 결과에서 입력받은 입출금 할 금액의 부호에 따라 입금할지 출금할지 나뉘어지도록 해야 할 것이다. 이를 입출금 연산을 해주는 함수 내에서 `if_else` 문을 이용하여 입출금 할 금액의 부호가 양수이면 입금하는 방향으로, 입출금 할 금액의 부호가 음수이면 출금하는 방향으로 진행되어야 할 것이다. 어느 방향으로 진행되어 잔액 값에 연산을 한 값을 저장하고, 입금 or 출금할 금액과 잔액을 출력하면서 프로그램을 마무리하도록 하자.

2. 프로그램 소스 + 각 라인 주석 설명

(Line 1 ~ Line 34)

```
Source.c
Week10 MID Report 2
(Global Scope)
main(void)

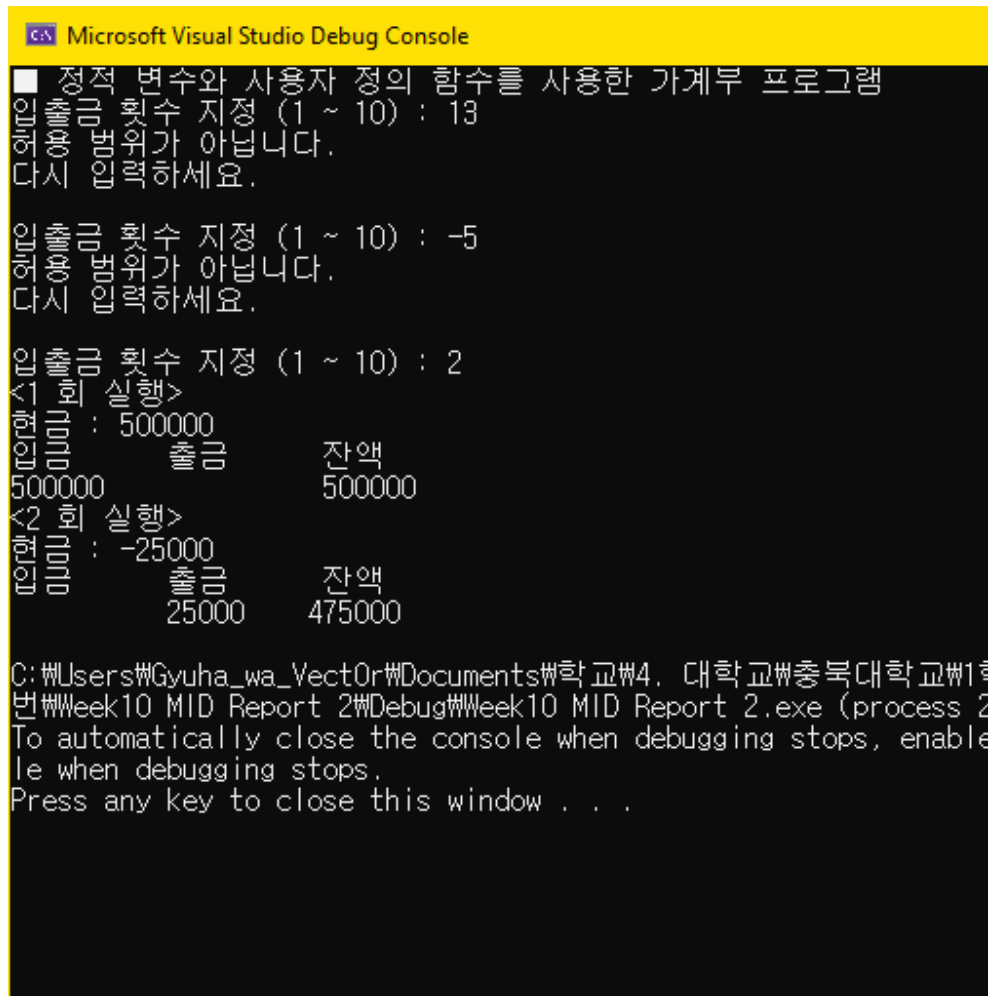
1 // 정적 변수와 사용자 정의 함수를 사용한 가계부 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3
4 // 매개변수만 구성 요소로 가지는 사용자 정의 함수 1개 정의
5 void accountBook(int a, int b, int c) { // 입출금 연산을 해주는 함수 accountBook
6     // 아래 main 함수에서 선언된 number, i, money를 정수형 매개변수 a, b, c로 받음
7
8     // 사용자 정의 함수에서 정적 지역 변수 1개 선언하여 입출금 내역 연산하여 출력
9     static int result; // 입출금 연산 후 남은 잔액을 저장하는 정수형 정적 지역변수 result 선언
10    // 1회 이상 입력 횟수 이하 동안 입출금 연산을 반복하여 진행되므로,
11    // 남은 잔액인 result의 값이 반복문의 매 턴마다 초기화 되면 안되므로 정적지역변수로 선언
12
13    // for문으로 입력 횟수에 따라 반복 수행
14    for (b; b <= a; b++) { // 메인함수의 i 값이 입출금 연산을 해주는 함수 accountBook에서는 b 값이 되고 이는 1로 초기화 되어있고,
15        // 메인함수에서 입력받은 입출금 횟수가 입출금 연산을 해주는 함수 accountBook에서는 a값이 된다.
16        // b를 1부터 1씩 증가시켜 a 값까지 도달 할 때 까지 아래의 내용을 반복
17        printf("<#> 회 실행>\n", b); // 입출금 할 횟수의 각 턴에 해당하는 수 출력
18        printf("현금 : "); // 입출금 할 금액을 입력해라는 메시지 출력
19        scanf_s("%d", &c); // 메인함수에서 0으로 초기화된 정수형 변수 money이 입출금 연산을 해주는 함수 accountBook에서는 c값이 되고,
20        // 입출금 할 금액을 입력받아 정수형 지역변수 c에 저장
21
22        if (c > 0) { // 만약에 입력 받은 입출금 할 금액이 양수이면
23            result = result + c; // 결과 값에 입력한 금액을 더함
24            printf("%s%10s%10s\n", "입금", "출금", "잔액"); // 입출금 연산 결과를 구성하는 UI를 실행화면에 맞게 공백을 넣어 구성
25            printf("%d%20d\n", c, result); // 입금할 금액과 입출금 연산하여 나오게되는 잔액을 출력
26        }
27        else if (c < 0) { // 만약에 입력 받은 입출금 할 금액이 음수이면
28            result = result + c; // 결과 값에 입력한 금액을 더함 (입력 받은 값 자체가 음수)
29            printf("%s%10s%10s\n", "입금", "출금", "잔액"); // 입출금 연산 결과를 구성하는 UI 구성
30            printf("%15d%10d\n", c * -1, result); // 출금할 금액과 입출금 연산하여 나오게되는 잔액을 출력
31        }
32    }
33 }
34 }
```

(Line 36 ~ Line 65)

```
Source.c
Week10 MID Report 2
(Global Scope)
main(void)

36 int main(void) { // main function 시작
37
38     // 메인 함수에서는 정수형 변수 3개 선언
39     int number; // 입력받은 입출금 횟수를 저장 할 정수형 변수 number 선언
40     int i = 1; // 입출금 연산을 해주는 함수 accountBook 내에서 입력받은 입출금 횟수에 따른 반복문을 돌리기 위한
41     // 정수형 변수 i 선언 후 1로 초기화 (입출금 횟수 : 1 ~ 10)
42     int money = 0; // 입출금 연산을 해주는 함수 accountBook 내에서 입출금 할 금액을 입력받고
43     // 연산에 사용되는 정수형 변수 money 선언 후 0으로 초기화
44     // 어짜피 money에 실제 다른 숫자가 들어올 것이니 초기화 해도 무방
45
46     printf("■ 정적 변수와 사용자 정의 함수를 사용한 가계부 프로그램\n"); // 프로그램 설명문 출력
47
48     comeBack: // 입출금 횟수 1 ~ 10 이 아닌 다른 숫자를 입력했을때 돌아오는 goto문의 체크포인트 comeBack 마련
49     printf("입출금 횟수 지정 (1 ~ 10) : "); // 사용자에게 입출금 할 횟수를 입력해라는 메시지 출력
50     scanf_s("%d", &number); // 입력받은 입출금 횟수를 저장할 정수형 변수 number에 저장
51
52     //가계부 입력 횟수(1~10)는 if_else문으로 허용 범위 판별
53     if (number > 0 && number < 11) { // 만약 입력받은 입출금 횟수가 1 ~ 10 내라면
54         accountBook(number, i, money); // main함수에서 선언된 정수형 변수 number, i, money를
55         // 입출금 연산을 해주는 함수 accountBook의 정수형 매개변수 a, b, c에 각각 보내버림
56     }
57
58     else { // 만약 입력받은 입출금 횟수가 1 ~ 10 내가 아니라면
59         printf("허용 범위가 아닙니다.\n다시 입력하세요.\n\n"); // 사용자에게 올바른 입출금 횟수를 입력해라는 메시지 출력
60         // 입력 횟수 허용 범위가 아닐 경우 goto 문 사용하여 다시 입력
61         goto comeBack; // 앞서 setting 한 체크포인트 comeBack로 다시 돌아가서 재실행
62     }
63
64     return 0; // main function 이 0으로 반환
65 }
```


3. 실행화면 캡처



```
Microsoft Visual Studio Debug Console
정적 변수와 사용자 정의 함수를 사용한 가계부 프로그램
입출금 횟수 지정 (1 ~ 10) : 13
허용 범위가 아닙니다.
다시 입력하세요.

입출금 횟수 지정 (1 ~ 10) : -5
허용 범위가 아닙니다.
다시 입력하세요.

입출금 횟수 지정 (1 ~ 10) : 2
<1 회 실행>
입출금 : 500000
잔액 : 500000
출금 : 500000
<2 회 실행>
입출금 : -25000
잔액 : 475000
출금 : 25000

C:\Users\Gyuha_wa_Vect0r\Documents\학교\4. 대학교\충북대학교\1학기\Week10 MID Report 2\Debug\Week10 MID Report 2.exe (process 2)
To automatically close the console when debugging stops, enable
the option 'Automatically close console when debugging stops' in
the 'Debug' menu.
Press any key to close this window . . .
```

4. 프로그램 동작 설명

출력 예시에 맞춰 동작 설명을 해 보겠다.

메인 함수에서 입출금 횟수를 입력 받는데 13 을 받아서 number 에 저장한다. 그러나 if_else 조건에 따라 goto 문을 통해 다시 위로 올라가게 된다. 그리고 다시 입출금 횟수를 입력 받는데 -5 를 받아서 number 에 저장한다. 이 또한 if_else 조건에 따라 goto 문을 통해 또 다시 위로 올라가게 된다. 그리고 또 다시 입출금 횟수를 입력 받는데 2 를 받아서 number 에 저장한다. 이는 if_else 조건에 성립하므로 입출금 연산을 해주는 함수 accountBook 을 불러오게 된다. 입력받은 입출금 횟수만큼 반복문을 돌리게 되는데, 입출금 할 첫 값은 500000 이고 이는 양수이므로 입금되어야 할 것이다. 따라서 연산이 끝난 후에

입금된 값과 연산 후의 잔액 값을 출력해준다. 잔액값은 정적 지역변수이므로 매 반복문이 돌때마다 값이 초기화 되는 것이 아니라 계속 그 값으로 유지되기 때문에 다음 턴에서 입출금 연산을 하여 잔액을 표현할 때, 전 턴에서 나왔던 잔액을 바탕으로 연산하여 출력이 될 것이다. 자 그러면 1 회 실행이 끝난 후 2 회 실행으로 넘어가게 되는데 이때 입력 받는 입출금 할 값은 -25000 원이고 이는 음수이므로 출금되어야 할 것이다. 따라서 연산이 끝난 후에 출금된 값과 연산 후의 잔액 값을 출력해준다. 반복문은 끝이 났고, 이 사용자 정의함수는 void 형이므로 리턴값이 따로 없다. 따라서 프로그램을 끝낸다.

요약 정리 문제

1. 변수, 자료형, 수식, 연산자

변수 (Variable) : 프로그램에서 일시적으로 데이터를 저장하는 공간이며 저장된 값의 변경이 가능한 공간

상수 (Constant) : 저장된 값의 변경이 불가능한 공간 (ex : 3.14, 100, 'A', "Hello World!")

자료형 :

자료형			설명	바이트수	범위
정수형	부호있음	short int	short형 정수	2	-32768~32767
		int	정수	4	-2147483648~2147483647
		long	long형 정수	4	-2147483648~2147483647
	부호없음	unsigned short	부호없는 short형 정수	2	0~65535
		unsigned int	부호없는 정수	4	0~4294967295
		unsigned long	부호없는 long형 정수	4	0~4294967295
문자형	부호있음	char	문자 및 정수	1	-128~127
	부호없음	unsigned char	문자 및 부호없는 정수	1	0~255
부동 소수점형		float	단일정밀도 부동 소수점	4	$1.2 \times 10^{-38} \sim 3.4 \times 10^{38}$
		double	두배정밀도 부동 소수점	8	$2.2 \times 10^{-308} \sim 1.8 \times 10^{308}$

수식 (Expression) : 변수, 상수, 연산자의 조합, 연산자와 피연산자로 나누어짐

연산자 : 기본적으로 좌측에서 우측방향으로 이동하면서 연산이 이루어짐.

산술 연산자	사용법	의미
+	a+b	a의 값과 b의 값을 더함
-	a-b	a의 값에서 b의 값을 뺌
	-a	a에 저장된 값의 부호를 바꿈
*	a*b	a의 값과 b의 값을 곱함
/	a/b	a의 값을 b의 값으로 나눔
%	a%b	a의 값을 b의 값으로 나눈 나머지

대입 연산자	사용 예	연산과 의미
=	a=b=0;	b=0; a=0;
+=	a+=b;	a=a+b;
-=	a-=b;	a=a-b;
=	a=b;	a=a*b;
/=	a/=b;	a=a/b;
%=	a%=b;	a=a%b;

논리 연산자	사용법	의미
&&	a&& b	a에 저장된 값과 b에 저장된 값의 논리곱(AND)
	a b	a에 저장된 값과 b에 저장된 값의 논리합(OR)
!	!a	a에 저장된 값에 대한 부정(NOT)

관계 연산자	사용법	의미	결과값
<	a<b	a에 저장된 값이 b의 값보다 작은가?	참이면 1, 거짓이면 0
<=	a<=b	a에 저장된 값이 b의 값보다 작거나 같은가?	
>	a>b	a에 저장된 값이 b의 값보다 큰가?	
>=	a>=b	a에 저장된 값이 b의 값보다 크거나 같은가?	
==	a==b	a에 저장된 값이 b의 값과 같은가?	
!=	a!=b	a에 저장된 값이 b의 값과 다른가?	

조건 연산자	사용법	의미
?:	수식1 ? 수식2 : 수식3	수식1의 연산 결과가 참(1)이면 수식2를 처리 수식1의 연산 결과가 거짓(0)이면 수식3을 처리
	k=(a>b) ? a : b	a가 b보다 크면 k에 a값을 저장 그렇지 않으면 k에 b값을 저장

coma 연산자 사용법	처리 순서
int i=10, b=5;	int i=10;
	int b=5;
x = (y+=3, y+15);	y=y+3;
	x=y+15;

++ 연산자		-- 연산자	
연산식	의미	연산식	의미
a++;	a=a+1;	a--;	a=a-1;
++a;		--a;	

캐스트 연산자 사용법	의미
(float) a / 3.14;	변수 a를 float형으로 변환하여 나눗셈 처리
(int) (a/b);	a/b의 연산 결과를 int형으로 변환

2. 조건문

if 문

```
if (조건식)
    조건식이 참인 경우에 처리할 문장
```

if~else 문

```
if (조건식1)
    조건식1이 참인 경우에 처리할 문장
else if(조건식2)
    조건식2가 참인 경우에 처리할 문장
else
    조건식 모두 거짓인 경우에 처리할 문장
```

중첩 if 문

```
if (조건식1)
    if (조건식2)
        조건식1도 참이고 조건식2도 참인 경우에 처리할 문장
```

switch case,
default, break 문

```
switch (value)
{
    case 1 : 입력 값(value)이 1일 때 처리할 문장;
        break;
    case 2 : 입력 값(value)이 2일 때 처리할 문장;
        break;
    case 3 : 입력 값(value)이 3일 때 처리할 문장;
        break;
    default : 입력 값(value)과 일치하게 없을 때 처리할 문장;
}
```

continue 문 : 반복을 중단시키지 않고 다음 반복이 이루어지도록 함.

go to 문 : 프로그램의 실행 순서를 특정한 곳으로 이동(jump).

3. 반복문 : 어떤 조건이 만족될 때까지 루프를 도는 구조

for 문

```
for(초기식 ; 조건식 ; 증감식)
    조건식이 참인 경우에 반복할 문장;

정해진 횟수만큼 반복하는 구조
```

중첩된 for문

```
for(초기식 ; 조건식1 ; 증감식)
{
    // 조건식1이 참일 경우 안에 있는 조건식2를 실행
    for(초기식 ; 조건식2 ; 증감식)
        조건식2가 참일 경우 반복할 문장
}
```

반복문 안에 다른 반복문이 위치

while문

```
while(조건식)
{
    조건식이 참일 경우에 반복할 문장;
}
```

주어진 조건이 만족되는 동안 문장들을 반복 실행한다.

-do while문

```
do //조건이 거짓이라도 1번은 반복처리하는 것이 특징
{
    반복할 문장;
} while(조건식);
```

적어도 한번은 반복문장을 실행한다.

4. 함수와 변수

1. 함수

사용자가 정의하는 함수를 사용하는 프로그램의 구성과 형식

- 함수원형(prototype)의 선언 : 사용자 정의 함수를 사용하겠다는 것을 컴파일러에게 미리 알려주는 역할
- 함수정의(definition)부분 : 함수가 처리해야할 기능들을 구체적으로 작성
(함수의 결과 값(반환 값)이 있는 경우에는 return문과 함께 변수 또는 연산식 사용.

함수 원형의 선언	데이터 형 함수이름 (함수인자들);
함수 정의	데이터 형 함수이름 (함수인자들); { 변수선언; 문장; [return 변수 또는 수식;] }

2. 변수

- 지역(local)변수와 전역(global)변수

변수 종류	의미	사용방법
지역변수	그 변수의 사용 범위가 어떤 한 함수 안에서만 사용	함수 또는 복합문 안에서 변수를 선언
전역변수	범위의 제한 없이 프로그램의 모든 함수에서 접근가능	함수 밖에서 변수를 선언

☞ 동(auto)변수

- : 함수 안에서 선언하는 지역 변수들은 모두 자동(auto) 변수에 해당.
- : 함수가 호출되어 실행될 때만 기억 공간이 확보(유지)되고 함수의 실행이 종료됨과 동시에 자동으로 소멸되는 변수
- : 자동 변수를 선언하기 위해서 auto라는 기억 클래스 지정자를 사용하지만 생략가능.

☞ 정적(static)변수

- : static 지정자를 사용
- : 프로그램이 종료될 때까지 변수의 수명이 지속
- : 초기화를 하지 않아도 0으로 초기화(한번 초기화된 정적 변수는 더 이상 초기화X)
- : 함수 내부에서 선언될 경우=내부 정적 변수
- : 함수 외부에서 선언될 경우=외부 정적 변수

☞ 외부(extern)변수

- : 어떤 모듈의 함수에서도 전역 변수로서 사용하고자 한다면 extern 지정자를 사용.
- : 파일 외부에서 선언된 전역 변수를 해당 프로그램 파일 내부에서 전역 변수로 사용할 경우 지정

☞ 레지스터(register)변수

- : 자동 변수와 기능이 동일하지만 CPU의 레지스터를 사용한다는 점에서 차이.
- : 크기에 제한이 있기 때문에 char형, int형으로만 사용
- : 프로그램의 실행속도를 증가시키기 위해 주로 반복문의 제어변수에 사용.