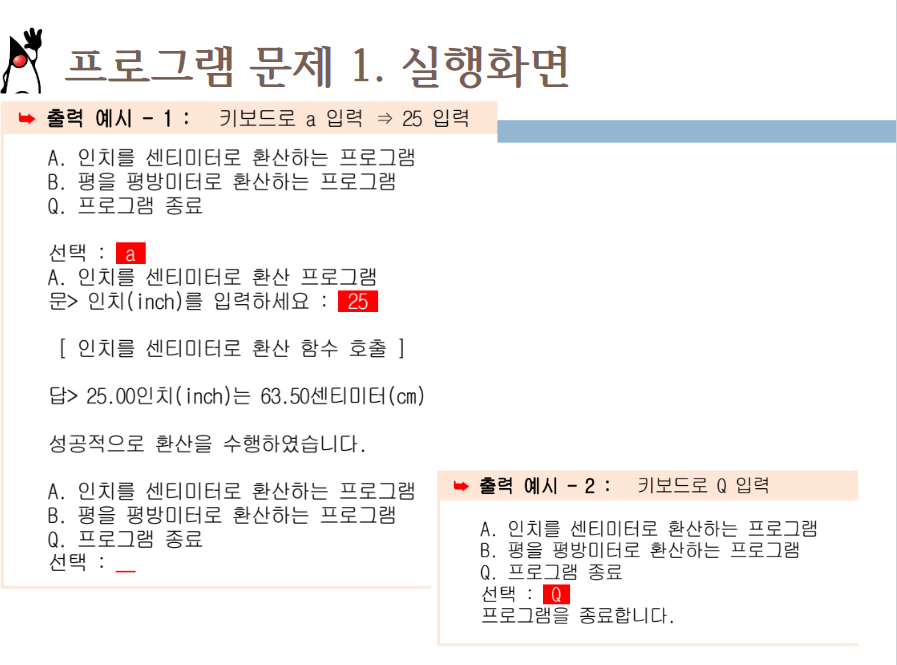
기초컴퓨터프로그래밍 10분반 25\_이규하 (2021039057)

10주차 **중간고사 대체** Report

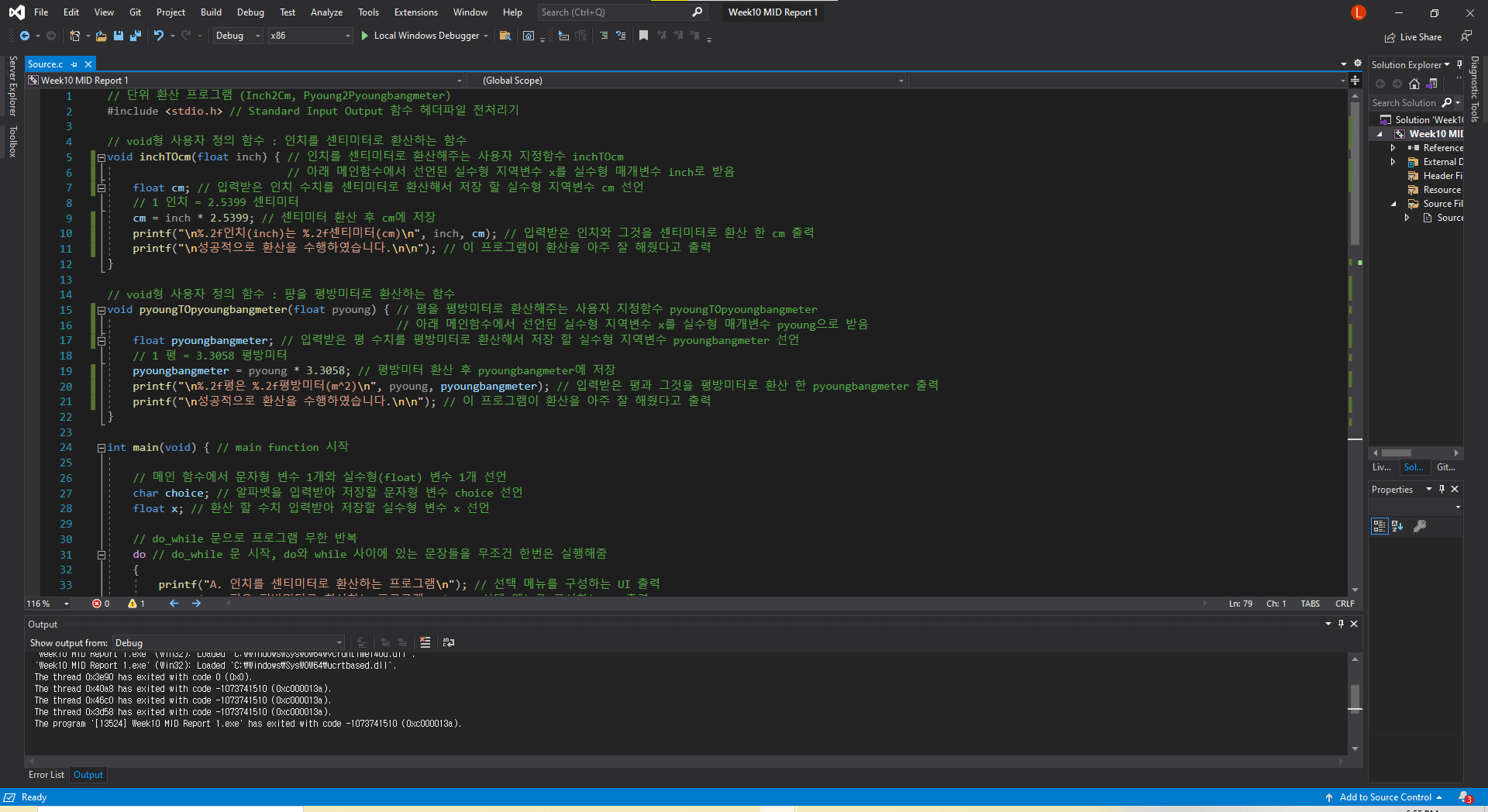
프로그램 문제 1) 다음 내용에 맞는 프로그램을 작성하시오.

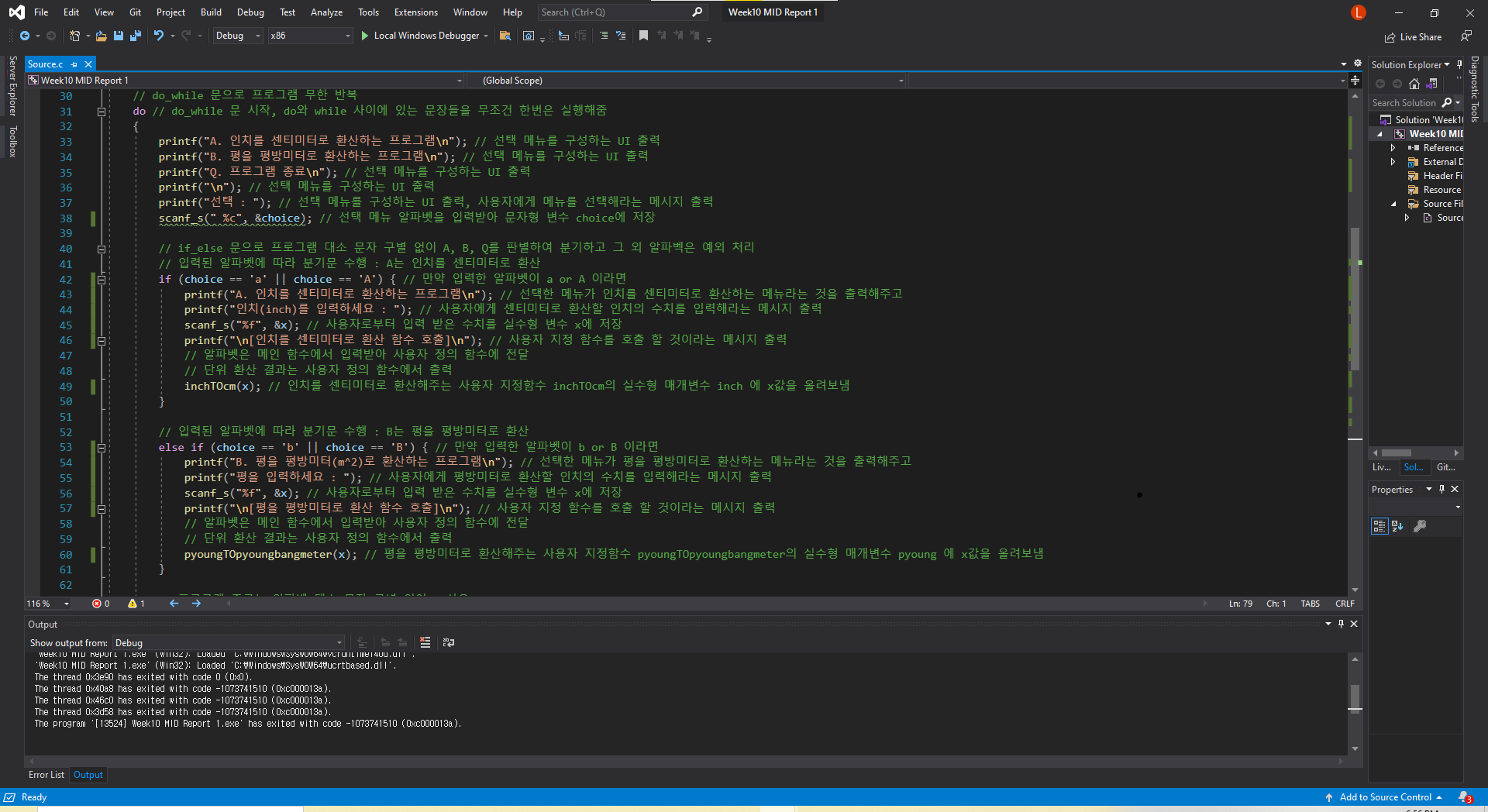
* void형 사용자 정의 함수: 인치를 센티미터로 환산하는 함수, 평을 평방미터로 환산하는 함수
* 1인치 = 2.5399센티미터, 1평 = 3.3058평방미터
* 메인 함수에서 문자형 변수 1개와 실수형(float) 변수 1개 선언
* do~while 문으로 프로그램 무한 반복
* if~else문으로 대소 문자 구별 없이 A, B, Q를 판별하여 분기하고 그 외 알파벳은 예외 처리
* A, B, Q 외의 알파벳 입력 시 continue 문 사용하여 다시 입력 요청
* 입력된 알파벳에 따라 분기문 수행: A는 인치를 센티미터로 환산, B는 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

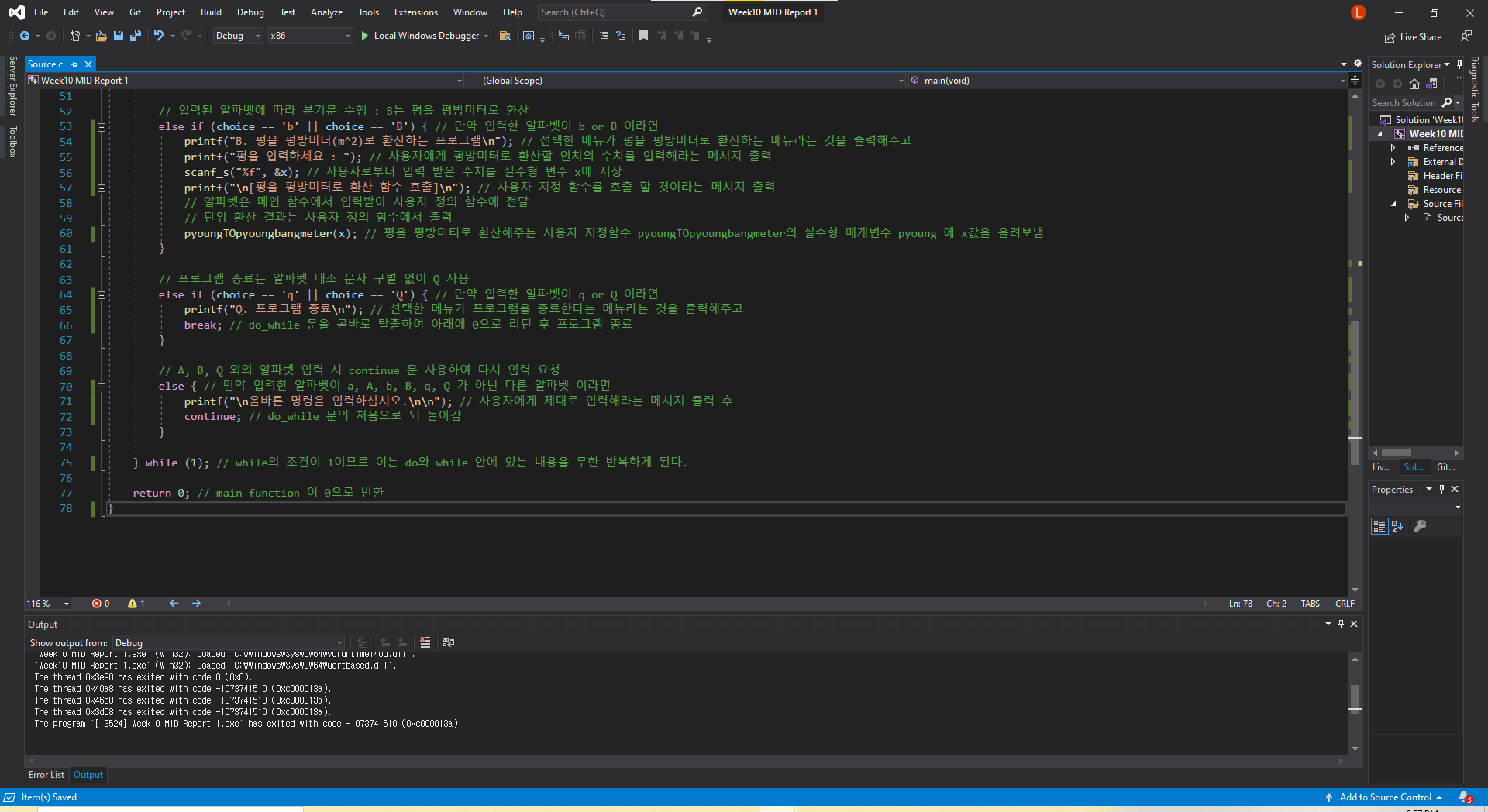
1. 문제 분석

단위 환산을 해주는 프로그램이다. 문제 조건에 따라 void형으로 인치를 센티미터로 환산해주거나, 평을 평방미터로 환산해주도록 하는 사용자 정의 함수를 구성하여 그 내에서 환산 연산(1인치 = 2.5399센티미터, 1평 = 3.3058평방미터)이 이루어 지도록 하고 환산 결과를 출력해주도록 하자. void 형 함수이기 때문에 리턴 값이 필요하지 않을 것이다. 그리고 문제조건에 따라 메인 함수 내에서 do\_while문을 통해 무한 반복을 해야 하므로 while의 조건식을 1로 설정하면 되겠다. 그리고 do와 while 사이에 인치를 센티미터로 환산하는 기능과 평을 평방미터로 환산하는 기능을 알파벳으로 선별할 수 있는 if문을 설정하여 사용자로부터 a 또는 A를 입력 받게 되면 인치를 센티미터로 환산하는 함수를 실행할 수 있도록 해주고, b 또는 B를 입력 받게 되면 평을 평방미터로 환산하는 함수를 실행할 수 있도록 해주자. 그리고 q 또는 Q를 입력 받게 되면 break을 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 continue를 이용해 다시 입력을 받게 해주자. 이런 알파벳을 입력 받아 저장할 문자형 변수를 선언해주고, 환산할 수치를 입력 받아 저장할 실수형 변수를 선언해주자.

1. 프로그램 소스 + 각 라인 주석 설명

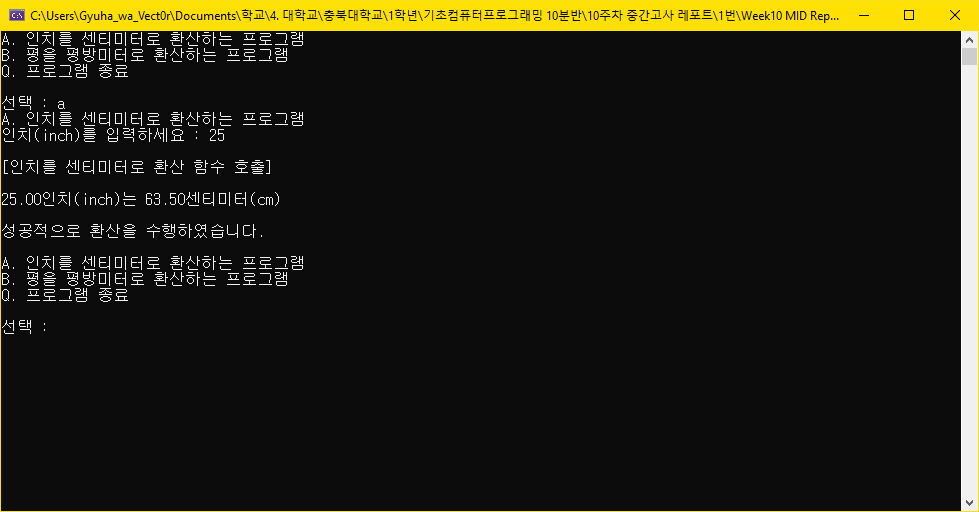
(Line 1 ~ Line 33)

(Line 30 ~ Line 61)

(Line 51 ~ Line 78)

1. 실행화면 캡처

(출력 예시 1: 키보드로 a 입력 -> 25 입력) (출력 예시 2: 키보드로 Q 입력)



1. 프로그램 동작 설명

출력 예시에 맞춰 동작 설명을 해 보겠다.

(출력 예시 1)

메인 함수에서 메뉴 선택 알파벳 a를입력받아 문자형 변수 choice에 저장한다.

choice == ‘a’ 이므로 if문의 조건에 따라 인치를 센티미터로 환산하는 쪽으로 진행되어야 한다. 사용자로부터 센티미터로 환산할 인치를 입력 받고, 위에서 마련한 사용자 지정 함수 inchTOcm의 매개변수로 그 입력 받은 것을 넘겨준다. 사용자 정의 함수 inchTOcm에서 입력 받은 인치 값에다 2.5399를 곱해주고 inchTOcm 내의 실수형 지역변수 cm에 저장해준다. 그리고 입력받은 인치 값과 환산한 센티미터 값을 출력해준다. while의 조건문이 1이므로 환산을 선택하는 메뉴를 제시해주면서 그곳에서 다시 시작한다.

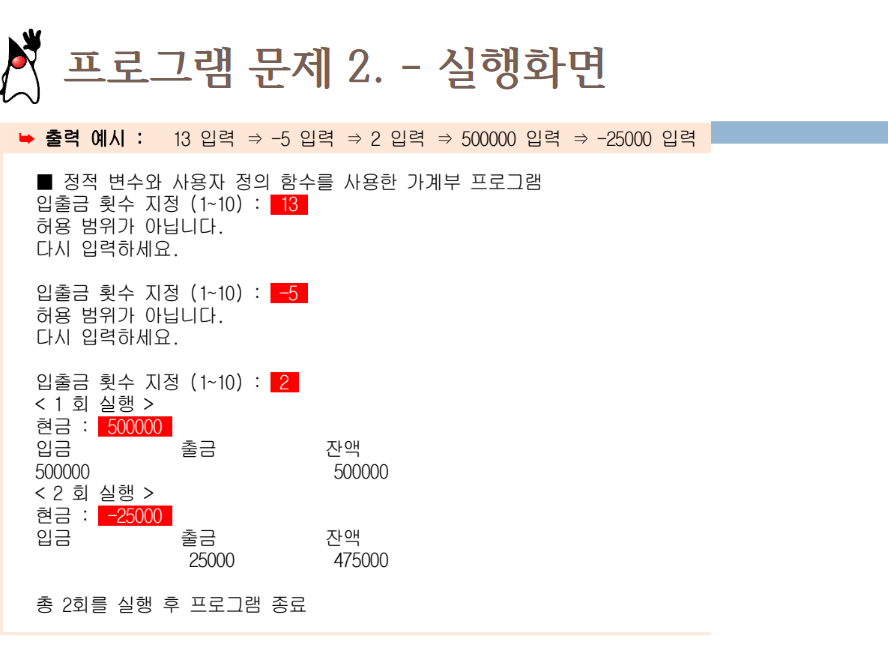
(출력 예시 2)

메인 함수에서 메뉴 선택 알파벳Q를입력받아 문자형 변수 choice에 저장한다.

choice == ‘Q’ 이므로 if문의 조건에 따라 프로그램을 종료하는 쪽으로 진행되어야 한다. 친절하게 프로그램을 종료한다는 메시지를 출력해주고 break 문에 의해 do\_while 문에서 빠져나가 메인 함수에 0으로 반환되면서 프로그램이 종료가 된다.

프로그램 문제 2) 다음 내용에 맞는 프로그램을 작성하시오.

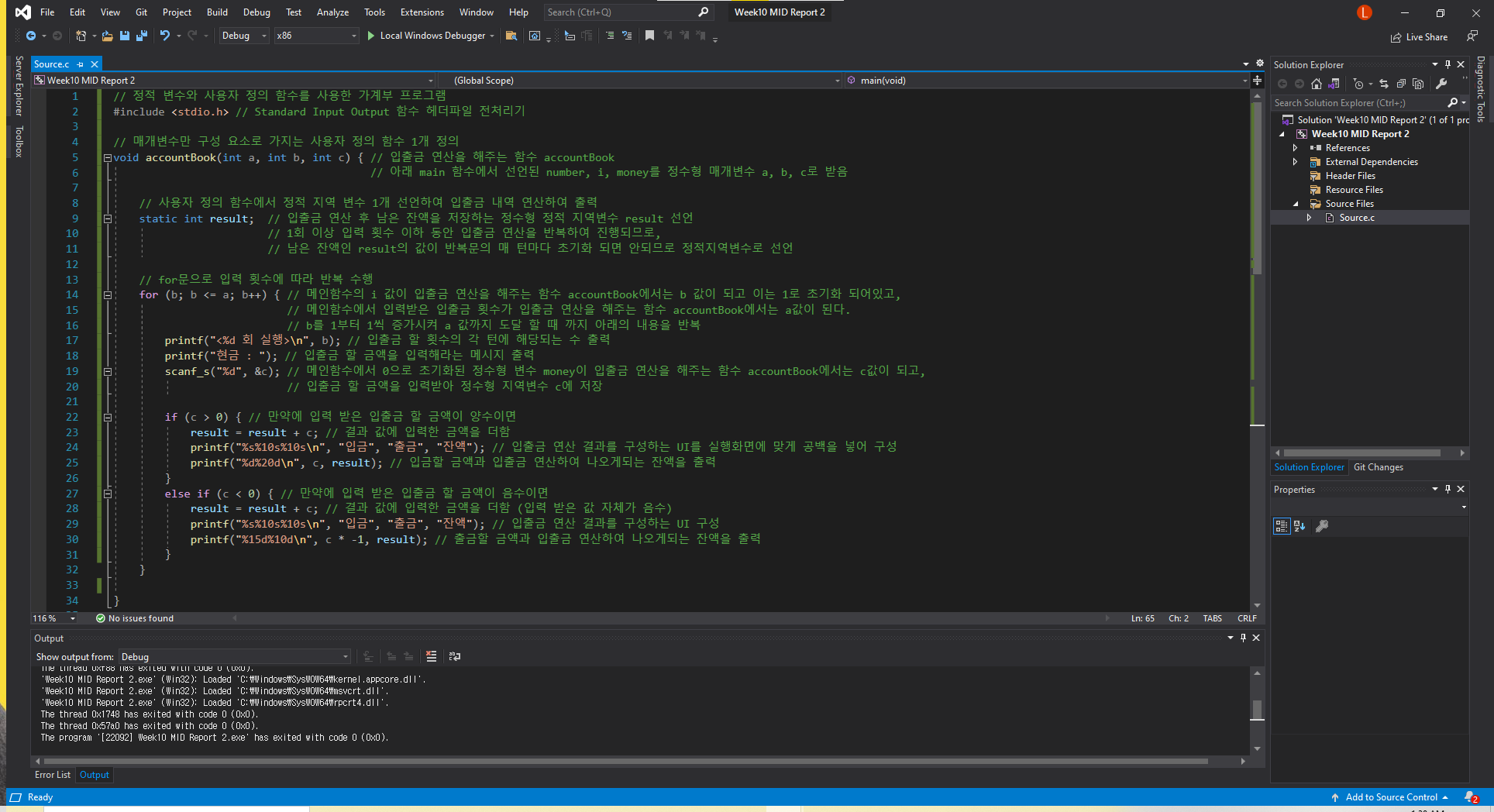
* 매개변수만 구성 요소로 가지는 사용자 정의 함수 1개 정의
* 메인 함수에서는 정수형 변수 3개 선언
* 가계부 입력 횟수 (1 ~ 10)는 if ~ else 문으로 허용 범위 판별
* 입력 횟수 허용 범위가 아닐 경우 goto 문 사용하여 다시 입력
* for문으로 입력 횟수에 따라 반복 수행
* 사용자 정의 함수에서 정적 지역 변수 1개 선언하여 입출금 내역 연산하여 출력
* 기타사항은 [출력 예시] 참조

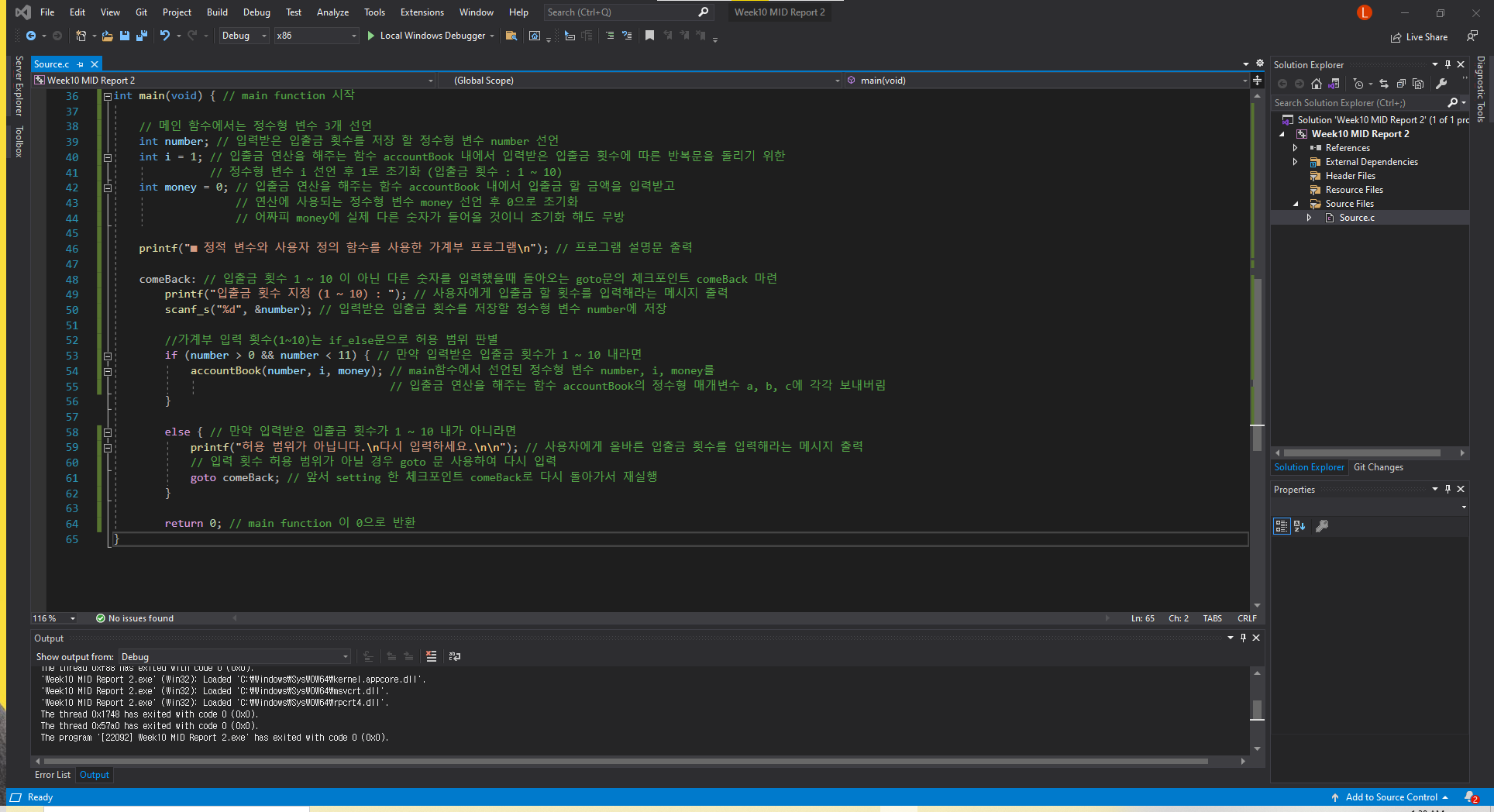


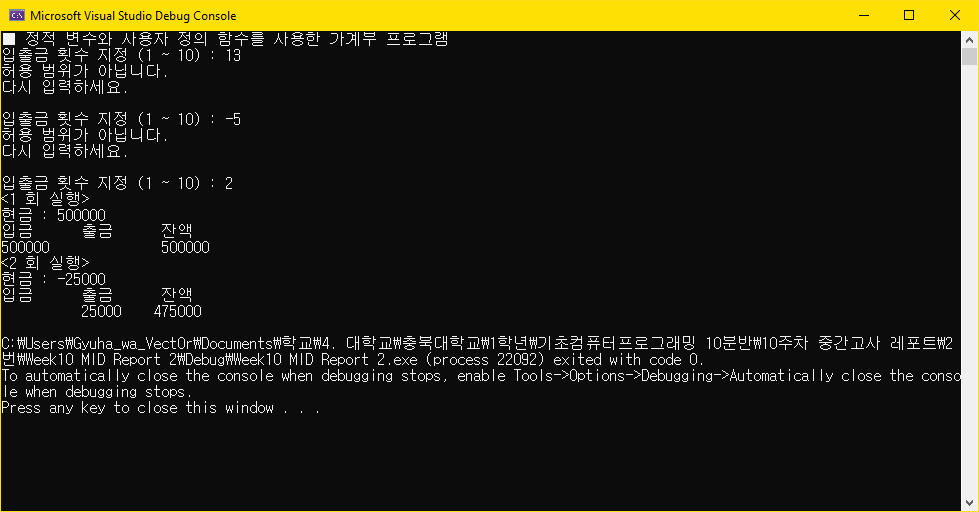
1. 문제 분석

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\_else문을 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 goto문을 통해 횟수를 다시 입력 받도록 하자. 이러한 내용을 메인함수에서 구현하기 위해서는 3개의 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 1부터 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.

1. 프로그램 소스 + 각 라인 주석 설명

(Line 1 ~ Line 34)

(Line 36 ~ Line 65)

1. 실행화면 캡처
2. 프로그램 동작 설명

출력 예시에 맞춰 동작 설명을 해 보겠다.

메인 함수에서 입출금 횟수를 입력 받는데 13을 받아서 number에 저장한다. 그러나 if\_else 조건에 따라goto문을 통해 다시 위로 올라가게 된다. 그리고 다시 입출금 횟수를 입력 받는데 -5를 받아서 number에 저장한다. 이 또한 if\_else 조건에 따라 goto 문을 통해 또 다시 위로 올라가게 된다. 그리고 또 다시 입출금 횟수를 입력 받는데 2를 받아서 number에 저장한다. 이는 if\_else 조건에 성립하므로 입출금 연산을 해주는 함수 accountBook을 불러오게 된다. 입력받은 입출금 횟수만큼 반복문을 돌리게 되는데, 입출금 할 첫 값은 500000이고 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 1회 실행이 끝난 후 2회 실행으로 넘어가게 되는데 이때 입력 받는 입출금 할 값은 -25000원이고 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

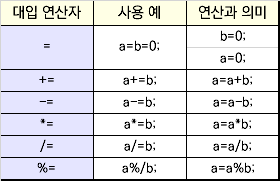
변수 (Variable) : 프로그램에서 일시적으로 데이터를 저장하는 공간이며 저장된 값의 변경이 가능한 공간

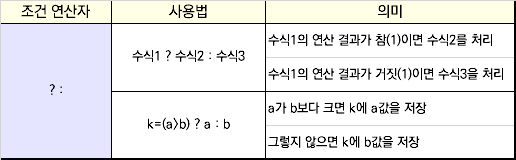
상수 (Constant) : 저장된 값의 변경이 불가능한 공간 (ex : 3.14, 100, ‘A’, “Hello World!”)

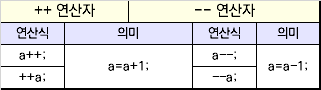
자료형 :

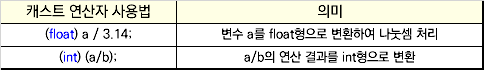
수식 (Expression) : 변수, 상수, 연산자의 조합, 연산자와 피연산자로 나누어짐

연산자 : 기본적으로 죄측에서 우측방향으로 이동하면서 연산이 이루어짐.









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, break문 | 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;  } |

continue문 : 반복을 중단시키지 않고 다음 반복이 이루어지도록 함.

go to문 : 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

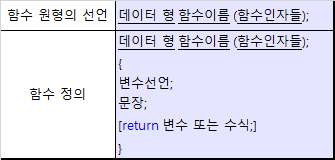
사용자가 정의하는 함수를 사용하는 프로그램의 구성과 형식

- 함수원형(prototype)의 선언 : 사용자 정의 함수를 사용하겠다는 것을 컴파일러에게

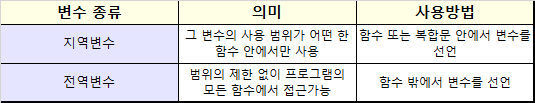
미리 알려주는 역할

- 함수정의(definition)부분 : 함수가 처리해야할 기능들을 구체적으로 작성

(함수의 결과 값(반환 값)이 있는 경우에는 return문과 함께 변수 또는 연산식 사용.



2. 변수

- 지역(local)변수와 전역(global)변수

* 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

: 어ᄄᅠᆫ 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.

: 크기에도 제한이 있기 때문에 char형, int형으로만 사용

: 프로그램의 실행속도를 증가시키기 위해 주로 반복문의 제어변수에 사용.