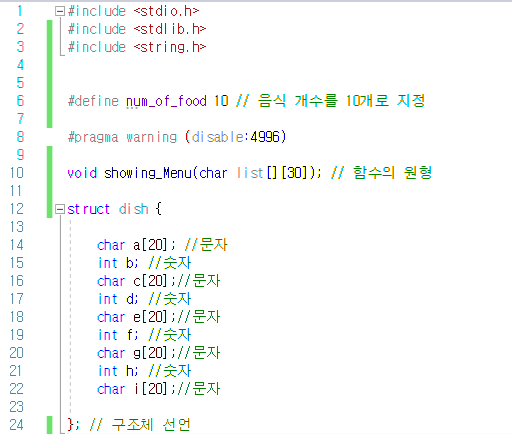
Project #2:: 프로그램 보고서:: 각 프로그램 기능에 대한 보고서

**(1) 사용한 각 함수에 대한 설명 (Input, Output, 알고리즘 설명)**

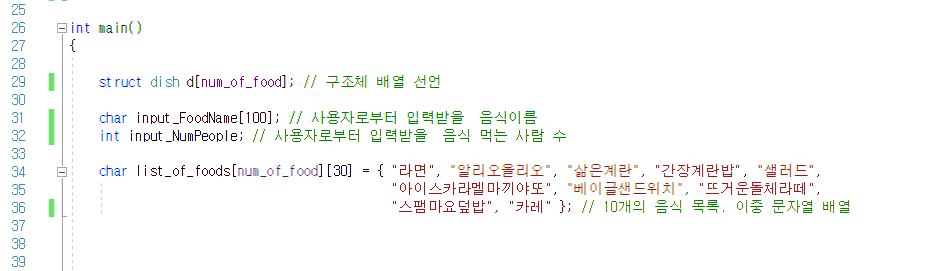
**< Food Recipe.cpp >**

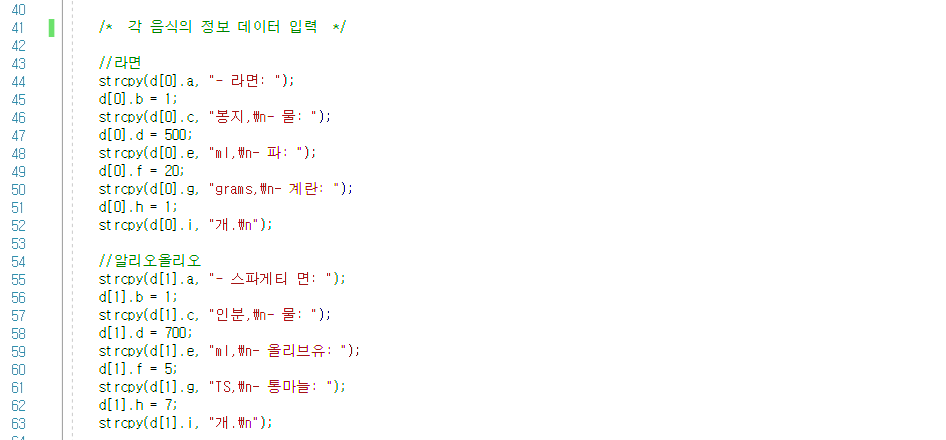


2번 프로그램의 맨 처음 부분이다. 제일 먼저, 이 프로그램을 작성하기 위해 필요한 헤더 파일들을 include시킨다. 그리고 음식의 개수를 10으로 지정해놓는다. 이렇게 define을 이용하면 나중에 음식의 개수를 바꾸고 싶을 때 여기의 숫자만 바꾸어주면 간편하게 일률적으로 값을 모두 수정시킬 수 있다. 프로그램의 가독성 또한 높아진다. 그 다음 줄은 앞으로 정의할 함수에 대한 원형을 써준 것이다.

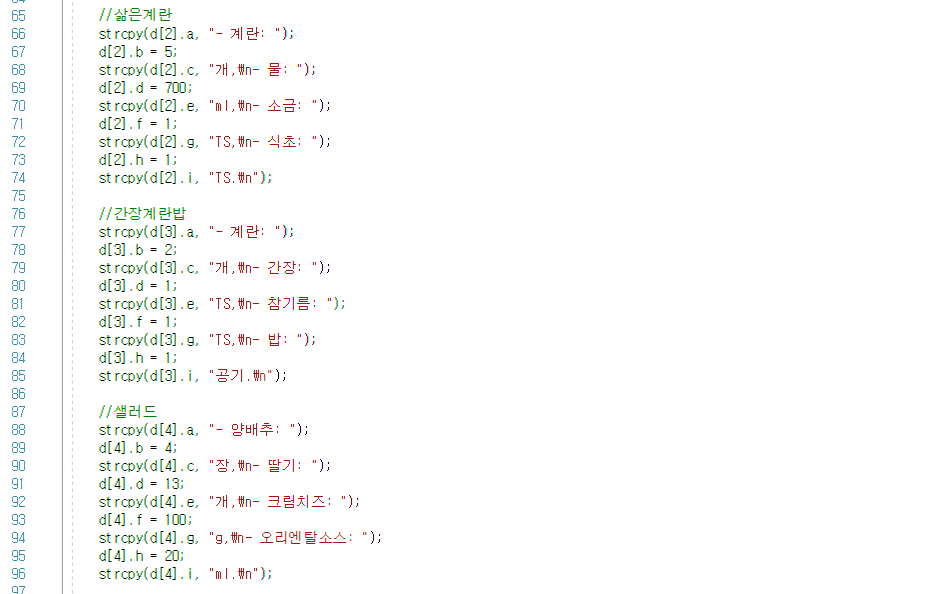
그 아래는 구조체를 선언한 것이다. 저 구조체 하나의 틀은 음식 한 가지에 대한 정보를 의미한다. 아직 구조체 변수를 선언하지는 않은 상태이다. 이 프로그램에서는 나중에 음식의 양에 먹는 사람의 수만큼 곱해서 출력시켜야 하기 때문에, 다음과 같이 여러 자료형을 번갈아 가면서 사용할 수 있는 구조체 개념을 이용하였다. 만약 사람 수만큼 곱해야 하는 처리를 하지 않아도 됐다면, 여러 자료형을 담을 수 있는 구조체 말고, 단순하게 문자열 이중배열을 사용했을 것이다. 이 부분에 대해서는 사연이 길다. 처음에는 문자열 이중배열을 사용하여 음식 재료들과 양의 정보들을 저장해두려고 했었다. 즉, 모두 문자열로 처리한다는 의미이다. 그래서 문자열에서 숫자만 추출하여 숫자로 변환시키고, 먹는 사람 수를 곱해주고, 다시 문자열로 수정시켜 처리하려 했었다. 이 길을 시도해 보았었는데 많이 험난하여 결국은 구조체 개념을 이용하게 되었다.

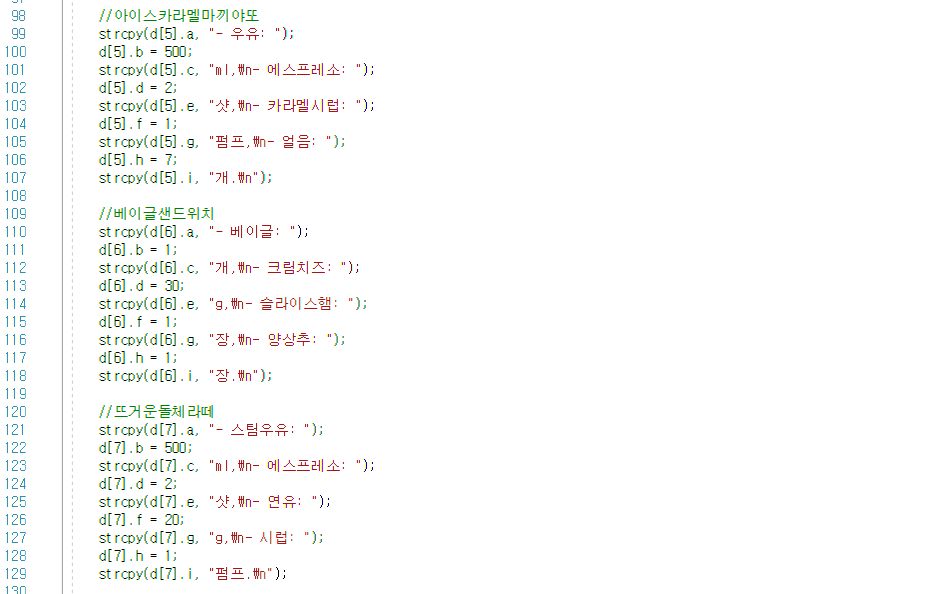
여기서부터는 메인 함수의 모습이다. (코드가 조금 길다)

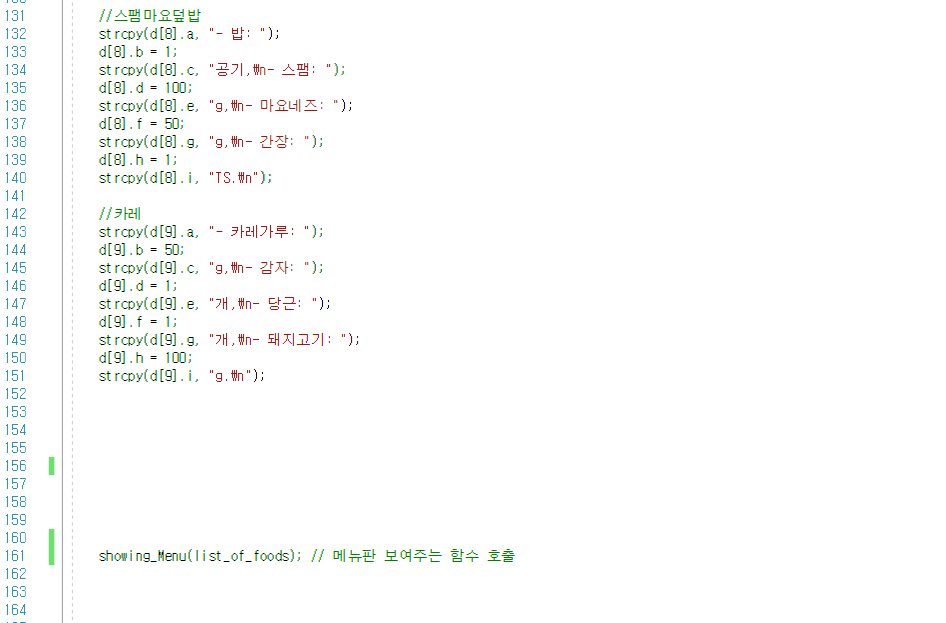




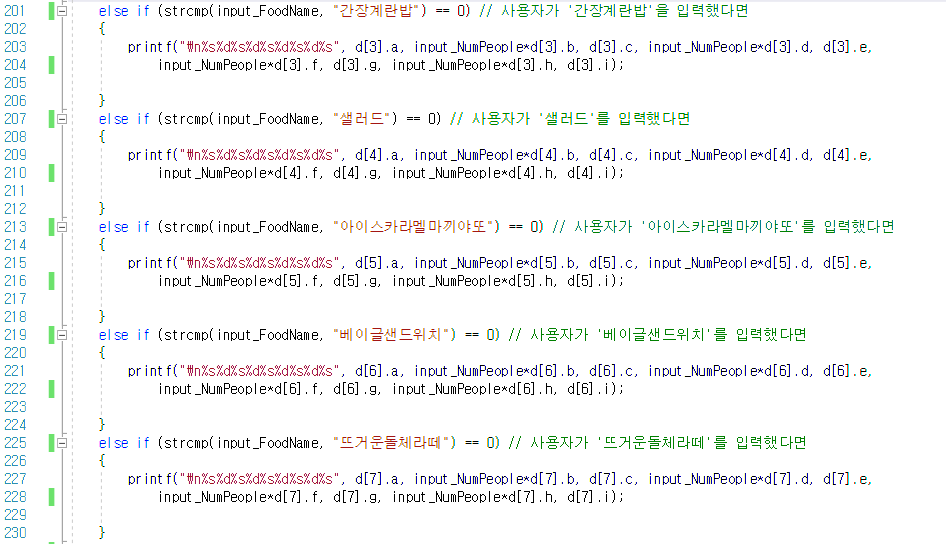
.

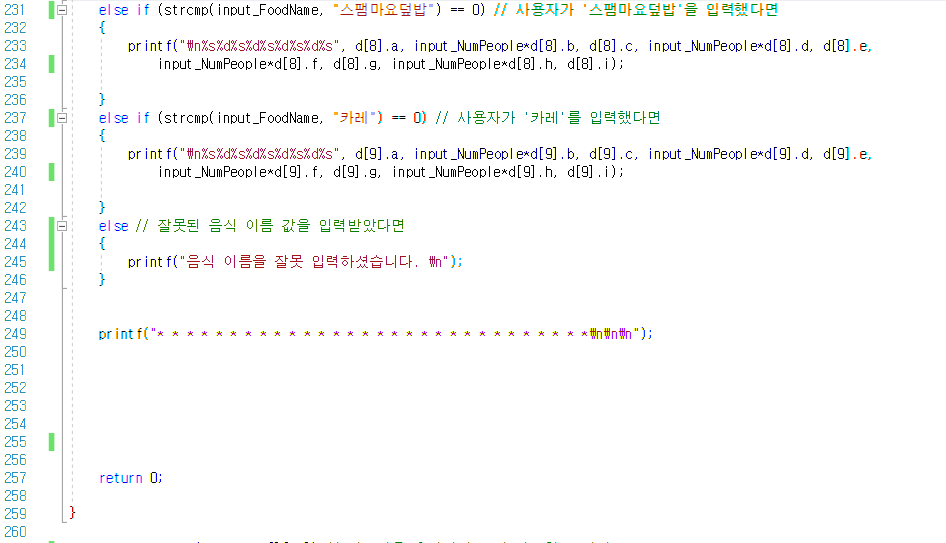












여기까지가 메인함수의 전체 모습이다. 메인 함수는 다음과 같은 구조 형태를 보인다.

***<*** *필요한 변수들 선언 -> 10개 음식에 대한 정보 데이터 입력 -> 함수 호출을 통한 사용자에게 메뉴판 보여주기 -> 사용자로부터 음식이름과 먹을 사람의 수를 입력 받기 -> 입력받은 음식에 대한 재료들과 그에 대한 양 정보를 출력하여 보여주기* ***>***

메인 함수의 맨 초반부에서는 필요한 변수들을 먼저 선언한다. 다음과 같은 변수들을 선언해주었다.

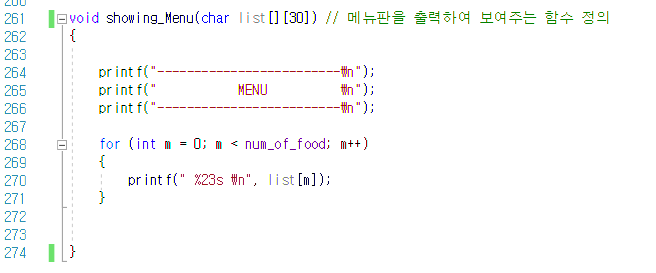
1. *struct dish d[num\_of\_food]; 🡪 음식 10개 각각에 대한 정보를 저장하기 위한 구조체 배열변수이다.*
2. *char input\_FoodName[100]; 🡪 사용자로부터 입력받을 ‘음식 이름’ 변수이다.*
3. *int input\_NumPeople; 🡪 사용자로부터 입력받을 ‘음식 먹을 사람의 수’ 변수이다.*
4. *char list\_of\_foods[num\_of\_food][30]; 🡪 문자열 이중배열이다. 음식이름 문자열 10개를 입력하여 초기화해준다. 행 하나 당 음식이름 문자열 하나를 보유하고 있다고 생각하면 된다. 메뉴판을 출력하여 보여줄 때 음식 목록들을 모두 출력해야 하는데 일일히 프린트 문으로 써줄 것이 아니라 for문을 돌려 출력할 것이다. 그 때 이 문자열 이중배열이 사용될 것이다.*

앞에서 음식 구조체 배열을 선언했었다. 이제 음식 10개에 대한 데이터를 모두 입력하여 준다. 이렇게 음식 데이터들을 미리 입력해두고, 사용자가 음식 이름을 입력 했을 때 저장해둔 데이터들 중 해당 음식에 대한 정보를 뽑아서 출력해줄 것이다. 사실 이렇게 하는 방식은 좋은 방식이 아닐 것이다. 왜냐하면 최근 수업 시간에도 언급 되었듯이, 이렇게 미리 입력시켜두는 것은 확장성이 없기 때문이다. 하지만 이 프로그램은 이렇게 할 수 밖에 없다. 말 그대로 사용자가 음식 데이터를 입력시켜주는 프로그램이 아니라, 저장해둔 데이터를 출력 시켜주는 프로그램이기 때문이다.

사용자가 음식 이름을 입력하기 전에, 메뉴에 어떤 음식들이 있는지 확인할 수 있도록 하기 위한 메뉴판도 만들었다. 메뉴판을 출력해주는 함수를 따로 만들었고, 여기 메인함수에서는 단순히 그 함수를 호출 시켰다.

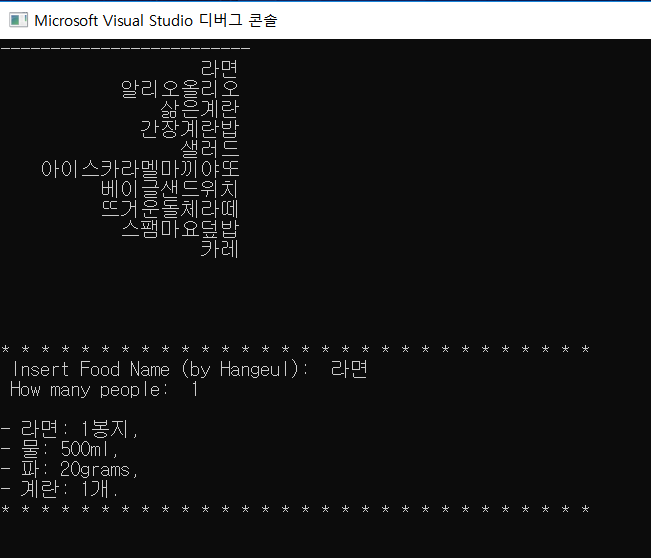
사용자로부터 음식 이름과 먹는 사람 수를 입력받도록 한다. 전자는 문자열 형태이고 후자는 int형의 숫자이다.

사용자로부터 음식 이름(문자열)을 입력받았다면, 해당 음식에 대한 정보를 문제에서 제시한 대로 출력시켜주어야 한다. 음식 종류는 다양하고 음식마다 진행되는 처리가 모두 다르기 때문에 if-else문을 사용하여 상황을 나누었다. 특히, if문을 쓸 때 주의해야 할 부분이 있었다. 단순하게 if ( input\_FoodName == ”라면” ) 으로 쓰면 안 된다. if ( strcmp(input\_FoodName, "라면") == 0 ) 이렇게 쓰도록 해야 한다. 이렇게 두 문자열을 비교함으로써 상황을 정의해야 한다. strcmp는 두 문자열을 비교해주는 함수이고, 두 문자열이 같을 때 0을 출력한다. 다양한 조건 속에서 한 곳이 충족되면 해당 if문의 처리가 진행된다. 해당 음식에 대한 재료들과 양을 출력시킬 때에는dish구조체의 멤버 변수 값들을 직접 접근하여 차례대로 출력시켰다. 먹는 사람 수만큼 음식 양을 곱해주는 처리도 여기서 깔끔하게 진행하였다. 이렇게 음식 이름을 입력 받아서, 그에 해당하는 if문에 대한 처리를 진행하고나면 메인 함수가 종료된다.

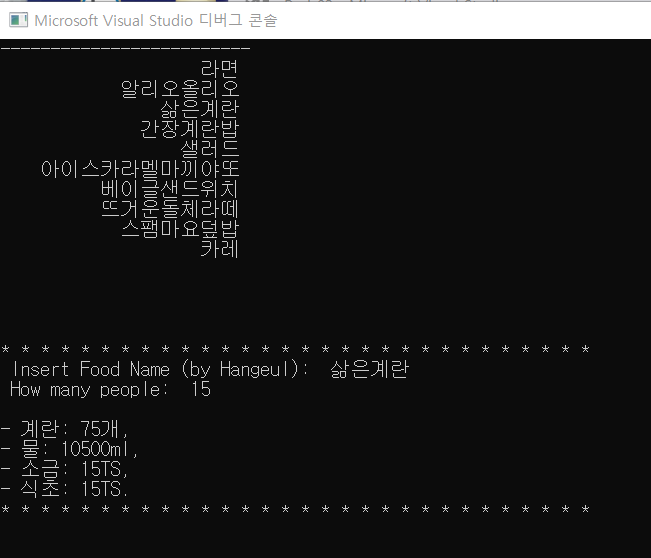


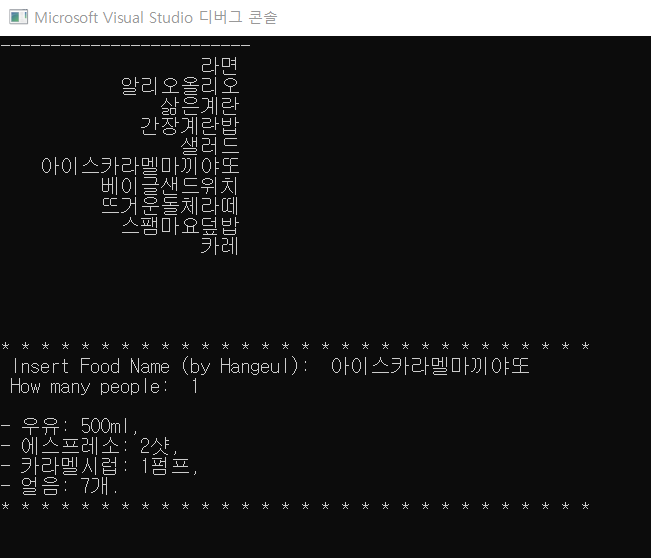
따로 정의한 함수는 하나 뿐이다. 위는 메뉴판을 출력해서 보여주는 함수인데, input으로는 문자열 이중 배열을 받는다. 즉, 메인 함수에서 호출할 때 인자로 문자열 이중 배열인 list\_of\_foods를 넘겨줄 것이다. 배열 이름 자체가 주소를 상징하기 때문에 상세하게 따지면 포인터 변수 즉, 배열의 주소를 넘겨준 것과 같다. 이 함수의 매개변수 char list[][30]는 문자열 이중 배열이라고 했는데, 행 하나당 음식이름 문자열 하나를 의미한다. 열을 30으로 설정한 이유는 음식이름 문자열의 길이가 충분히 길어질 수도 있기 때문에 그렇게 설정한 것이다. Output은 따로 없다. 말 그대로 메뉴판을 출력하여 보여주는 것이 목적인 함수이기 때문에 내보내는 리턴 값이 필요하지 않다.

**(2) 실행 결과 화면 단계별 Capture (작은 size 화면으로 Capture)**

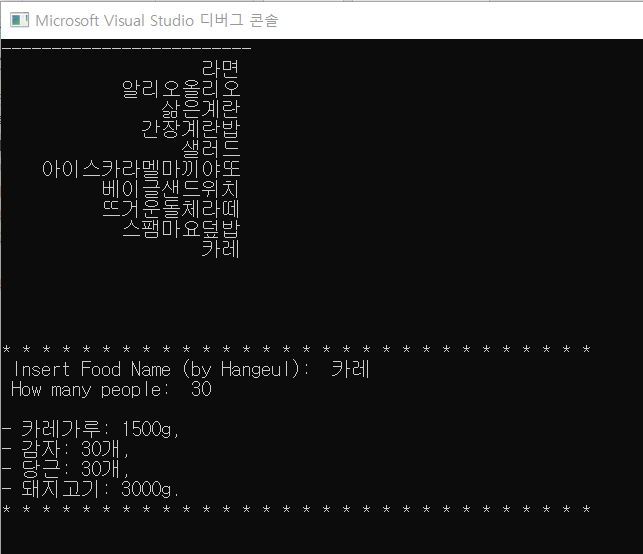




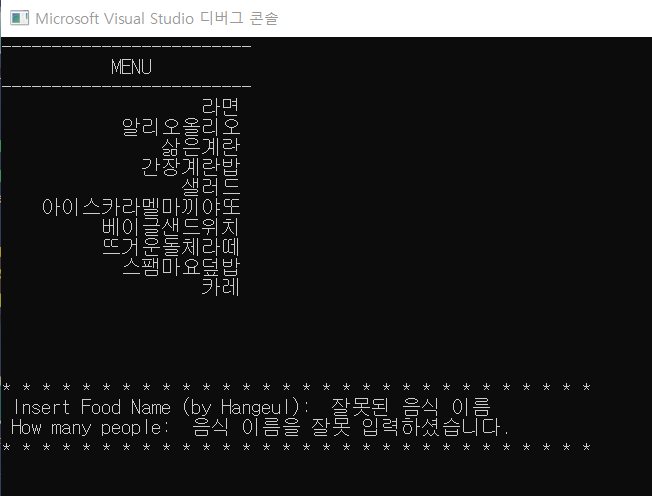








사용자가 제대로 된 음식 이름과 사람 수를 입력하면 문제에서 제시한 예시대로 출력된다. 음식의 양도 사람 수만큼 곱해져서 성공적으로 출력된다.





잘못된 음식 이름을 입력하면 위와 같이 출력된다. 또한 빈칸 공백을 이용하여 문자열을 입력했을 때에도 위와 같이 출력된다.