**2023 Spring OOP Assignment Report**

과제 번호 : 2

학번 : 20220124

이름 : 김문겸

Povis ID : kkomy

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.

독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.

문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

아래 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

1. **프로그램 개요**
   * 해당 프로그램은 덧셈(+), 뺄셈(-), 곱셈(\*), 나눗셈(/), 그리고 제곱근(sqrt)와 제곱(square)을 계산하는 프로그램이다. 프로그램이 실행되면, 연산자를 입력받은 후 숫자를 입력받아 연산자에 따른 계산 결과를 출력한다.
2. **프로그램의 구조 및 알고리즘**

* 해당 프로그램에서는 추가적으로 <cmath>와 <cstring> 헤더파일을 사용했다. cmath의 경우 제곱근을 계산하기 위해 불러왔고, cstring은 sqrt와 square 연산자를 인식하기 위한 함수 strcmp를 사용하기 위해 불러왔다.
* 해당 프로그램은 먼저 연산자를 입력받은 후, 숫자가 2개가 필요한 연산자(+,-,\*,/)와 1개가 필요한 연산자(sqrt, square)에 대한 경우를 분리하여 컴파일된다. 2개가 필요한 연산자는 숫자 2개를 입력받고 연산자에 따라 계산을 진행하여 결과값을 출력한다. 1개가 필요한 연산자의 경우 숫자 1개를 입력받고, 계산을 진행하여 결과값을 출력한다.
* 다음은 선언한 변수에 대한 설명이다.
  + - char oprt[10] : 연산자를 입력받기 위해 선언했다. +, -와 같이 한 개의 문자로 구성된 연산자 외에도 square과 같은 긴 문자열을 입력받는 연산자가 있기 때문에 넉넉한 크기의 배열을 선언했다.
    - float r1, r2 : 계산하고자 하는 수를 저장하는 변수들이다. 2개의 수를 필요로 하는 연산자ㅡ이 경우 둘 다 사용하지만, 한 개만 사용하는 sqrt, square의 경우 r1만을 사용한다.
* 다음은 전체적인 프로그램의 실행과정에 대한 설명이다.

처음 연산자를 입력받고, if문을 통해 숫자를 2개를 입력받을 것인지, 1개를 입력받을 것인 것 결정한다. 즉 해당 프로그램은 크게 두 if 조건문으로 분류된다.

|  |
| --- |
| if (oprt[1] == '\0') {  ~  }  else {  ~  } |

문자열은 항상 해당 문자열이 끝났다는 것을 표현해주기 위해 NULL(\0)을 마지막에 붙인다. 따라서 oprt[1] == ‘\0’이라는 것은 oprt[0]부터 들어가는 문자가 한 개로 끝났다는 것을 의미한다. 따라서 해당 if 조건문 내부에서는 r1, r2 2개를 입력받는다. 하지만 oprt[1]이 NULL이 아니라면 sqrt와 square 연산자를 의미하므로 r1 1개를 입력받는다.

<수 2개를 입력받는 연산자의 연산 및 출력 과정>

|  |
| --- |
| if (oprt[0] == '+') {  float result = r1 + r2;  cout << fixed;  cout.precision(result == (int)result ? 0 : 3);  cout << result;  } |

위 코드는 덧셈(+) 연산자에 대한 연산 및 출력을 담당하는 코드이다. 나머지 -, \*, / 또한 거의 유사한 코드의 구조를 지니고 있다. 먼저 연산 결과로 initialize된 변수 result를 선언한다. 이후 소수점에 대한 문제의 조건을 맞추기 위한 과정이 진행된다.

* + - 소수점 설정

문제에서는 계산 결과가 소수점 아래 3자리보다 길 경우 소수점 셋째 자리까지 반올림하여 출력하고, 만약 정수로 떠렁지는 경우에는 소수로 표현하지 않아도 된다는 세부조건이 주어져 있다. 우선 기본적으로 소수점을 조작하는 작업이기 때문에, 소수점을 고정시키는 기능을 담당하는 “cout << fixed;”를 사용한다. 이후 아래 코드에 cout.precision()을 사용했는데, cout.precision()은 소수의 전체 길이를 설정하는 cout의 member function이다. 괄호 안에는 전체 길이를 입력한다. 위 코드에는 괄호 안에 삼항연산자가 들어 있다. 위 코드의 삼항연산자는 만약 결과값인 result가 정수로 떨어진다면 result를 int형으로 표현했을 때 값이 동일할 것이다. 그렇다면 0을 반환하고, 그것이 아니라면, 즉 소수의 형태라면 3을 반환한다. 따라서 정수로 떨어지면 cout.precision(0), 소수점 아래 자릿수가 0개, 아니라면 cout.precision(3), 소수점 아래 자릿수가 3개로 출력이 설정된다. 이후 cout << result를 통해서 계산값을 출력한다.

<수 1개를 입력받는 연산자의 연산 및 출력 과정>

수 1개를 입력받고, 입력받은 연산자가 sqrt인지, square인지 확인한다. 이를 위해 strcmp를 사용한다. strcmp는 문자열을 비교하는데 사용하는 함수로, 두 문자열이 일치하면 0, 아니면 1을 반환한다. 우리는 일치하는 것을 확인하고 싶으므로 부정 논리 연산자 !를 앞에 붙여 if문의 조건으로 설정한다.

|  |
| --- |
| if(!strcmp(oprt,"sqrt")){  float result = sqrt(r1);  cout << fixed;  cout.precision(result == (int)result ? 0 : 3);  cout << result;  } |

위 코드는 sqrt에 대한 연산 및 출력 코드이다. 앞서 설명한 +,-,\*,/ 연산 코드와 똑같이 소수점 설정과 결과값 출력 코드로 구성되어 있다.

이렇게 계산 값을 출력하게 되면 최종적으로 프로그램이 종료된다.

1. **토론 및 개선**
   * + 일반적으로 잘 사용하지 않았던 삼항연산자를 사용하여 그 기능을 다시 한 번 상기시켰다. 원래 if 조건문을 사용하여 소수점을 설정하려고 했으나, 조금 더 소스코드를 간결하게 만드는 방법을 고민하던 중 삼항연산자를 떠올려서 이를 사용하였다.
     + 연산자를 char (array) 자료형으로 입력받으라는 조건이 있었기에 C언어 문자열 비교 함수인 strcmp를 사용하여야 했고, 이 때문에 추가적인 헤더파일 <cstring>을 사용하였지만, 조금 더 간단하고 실용적인 string을 사용하여 굳이 cstring을 사용하지 않고 간단하게 표현할 수 있을 것이다.
     + 각 연산자에 대한 조건문마다 result를 선언하지 않고, result를 main함수의 지역변수로 선언한 다음, 최종적으로 마지막에 result를 출력하는 형태로 작성했다면 더 간결한 코드가 되었을 것 같다.
2. **참고 문헌**