**2023 Spring OOP Assignment Report**

과제 번호 : 3

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 연산자와 2,8,16진수 2개를 입력받아서, 두 수를 연산자로 계산한 후 결과값을 10진수로 출력하는 프로그램이다. 이때 2, 8, 16진수를 구분하기 위해, 2진수의 앞에는 ‘0b’가 접두되고, 8진수의 앞에는 ‘0’이, 16진수의 앞에는 ‘0X’가 접두된다. 또한 연산자는 덧셈, 뺄셈, 곱셈, 나눗셈만을 받아들인다. 또한 입력 시 음수, 소수는 입력되지 않는 것으로 세부 조건을 명시했다.
2. **프로그램의 구조 및 알고리즘**

* 해당 프로그램은 연산자와 수 2개를 입력받는다. 이때 연산자는 문제 2와 달리 덧셈, 뺄셈, 곱셈, 나눗셈으로 제한되므로 반드시 2개의 수를 입력받아야 한다. 입력이 완료되면, 입력받은 수를 10진수로 변환하는 과정을 거치게 된다. 10진수로 변환이 완료되면, 연산자의 종류에 따라 계산을 진행하고, 결과값을 출력하게 된다.
* 해당 프로그램에서는 추가적으로 <string>과 <cmath> 헤더파일을 사용했다. string의 경우 string을 사용하기 위해 불러왔고, cmath의 경우 10진수로 변환하는데 필요한 n의 m제곱을 계산 할 수 있는 함수 pow를 사용하기 위해 불러왔다.
* 다음은 선언한 변수들에 대한 설명이다.
  + - char oprt[2] : 입력받은 연산자를 저장하기 위한 문자열이다.
    - string r1, r2 : 입력받는 2 or 8 or 16진수를 저장하기 위한 string이다. 문제 조건에서 string type으로 받으라는 조건이 존재한다.
    - double deci1 = 0, deci2 = 0 : r1과 r2를 10진수로 변환한 값을 저장하기 위한 변수로, 저장과정에서 r1, r2의 자릿수마다 덧셈과정이 필요해서 0으로 초기화를 진행했다.
* 다음은 전체적인 프로그램의 실행과정에 대한 설명이다.

연산자와 2/8/16진수 중 하나에 해당되는 수 2개가 입력된다. 이후 입력받은 수에 대해서 무슨 진수에 해당되는지 검토과정이 진행된다. 우선 2, 8, 16진수만 입력받기 때문에, r1, r2 string의 첫 칸은 0이 될 것이다.

|  |
| --- |
| if (r1[0] == '0') {  ~  } |

진수 판별의 첫번째 과정은 입력가능한 숫자인지 판별하는 것이다. 따라서 첫 칸이 무조건 0이어야 한다.(2진수 시작 : 0b, 8진수 시작 : 0, 16진수 : 0X)

|  |
| --- |
| if (r1[1] == 'b') {  int dgt = r1.length();  for (int i = 2; i < dgt; i++) {  int k = r1[i] - 48;  deci1 += k \* pow(2, dgt - 1 - i);  }  } |

위 코드는 2진수를 10진수로 변환하는 코드이다. r1[1]이 b라면 2진수에 해당하므로 조건문을 실행한다. 우선 변수 dgt를 선언하여 string인 r1의 길이를 r1.length()를 통해 구한다. 따라서 수가 시작되는 3번째 배열부터 마지막 배열까지 계산을 진행하여 10진수로 변환한다. r1[i]의 경우 char형으로서 숫자를 문자로 받아들이기 때문에 ASCII 코드 값을 인식한다. 따라서 0의 ASCII코드값인 48을 빼줌으로서 숫자로서의 값을 k에 저장한다. 이후 변환된 10진수를 나탄내는 변수인 deci1에 k 곱하기 2^(dgt -1 -i)를 더해준다. 이때 dgt -1 – i를 하는 이유는, dgt – i가 자릿수를 의미하고, 배열은 첫번째 배열이 0부터 시작하므로 이를 보정하기 위해 1을 빼주었다.

<10진수 변환>

* + - 2, 8, 16진수를 10진수로 변환하는 것은 간단하다. 각 자리수만큼 2/8/16의 거듭제곱을 곱하여 합하는 것이다. 예를들어 8진수 1201이 있다면, (1\*8^3 + 2\*8^2 + 0\*8^1 + 1\*8^0) 이런식으로 계산한다.

|  |
| --- |
| else if (r1[1] == 'X') {  int dgt = r1.length();  for (int i = 2; i < dgt; i++) {  int k;  if (r1[i] >= 65) {  k = r1[i] - 55;  }  else k = r1[i] - 48;  deci1 += k \* pow(16, dgt - 1 - i);  }  } |

위 코드는 16진수를 처리하는 코드이다. 16진수는 숫자인 1~9와 문자인 A~F로 구성되어 있기 때문에 ASCII 코드를 숫자값으로 변환하는 과정을 별도로 분리하여 계산하고, 마지막에 2진수와 똑같이 계산해주면 된다.

|  |
| --- |
| else {  int dgt = r1.length();  for (int i = 1; i < dgt; i++) {  int k = r1[i] - 48;  deci1 += k \* pow(8, dgt - 1 - i);  }  } |

위 코드는 8진수의 처리코드이다. 2, 16진수와 똑 같은 구조이다.

이 과정을 r2도 동일하게 거친다.

|  |
| --- |
| if (oprt[0] == '+') {  float result = deci1 + deci2;  cout << fixed;  cout.precision(result == (int)result ? 0 : 3);  cout << result;  } |

위 코드는 연산자 +에 대한 계산 처리 및 출력 코드이다. float형 변수 result를 선언하여 앞서 변환했던 10진수를 연산자에 따라 계산한 값을 저장하고, cout << fixed를 통해 소수점을 고정한 후 cout.precision()을 통해 소수점 아래 자릿수를 결정한다. 이때 괄호 안에 삼항연산자를 사용함으로써, 정수로 떨어진다면, 즉 result의 값이 result를 int형으로 변환한 값과 같다면 정수로만 표현해도 되므로 소수점 아래 자릿수를 0으로, 소수로 표현된다면 3자리까지 반올림하여 표현한다. 뺄셈, 곱셈, 나눗셈 또한 똑 같은 구조를 가지고 있다.

이후 cout << result를 통해 계산값이 출력된다

1. **토론 및 개선**

* 문제 1에서는 10진수를 2, 8, 16진수로 변환하는 프로그램에 대해서 다루었고, 문제 2에서는 계산기 프로그램을 다루었다. 이번 문제는 문제 1과 문제 2를 융합한 것으로, 문제 1의 변환 과정을 반대로 진행하는 것이 이번 프로그램의 키포인트였다.
  + - 현재 작성한 코드에서는 r1과 r2를 10진수로 변환하는 코드가 분리되어 작성되어 있다. 10진수로 변환하는 코드는 변수의 이름 말고는 완벽히 일치하는데, 변환하는 변수의 이름이 다르다고 똑같은 코드를 한번 더 써서 코드를 간결하지 못하게 만드는 것이 아쉽다. 이 점에 대해서 객체지향프로그래밍 언어인 C++을 좀 더 공부한다면 해결 할 수 있을 것이라 생각한다. 좀 더 공부하여 해결의 실마리를 찾고 싶다.

1. **참고 문헌**