**2023 Spring OOP Assignment Report**

과제 번호 : 1

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 10진수를 2, 8, 16진수로 변환하는 프로그램이다. 프로그램이 가장 처음 실행되면 10진수와 변환해야 할 진수를 int형으로 입력받게 된다. 이후 프로그램은 입력받은 값에 따라 10진수를 변화시켜 해당 값을 원하는 진수의 형태로 출력한다.
   * 문제에서는 C++ 상에서 쉽게 8진수와 16진수로 변환시킬 수 있는 std::hex, std::oct에 대해 사용을 금하고 있다. 따라서 우리가 일반적으로 진수 변환 시 사용하는 진법변환 알고리즘을 이용한다. 진법변환 알고리즘이란 10진수를 변환하고자 하는 진수의 n제곱씩 나누어줌으로써 변환된 진수의 값을 자리마다 구하는 방법을 의미한다.
2. **프로그램의 구조 및 알고리즘**

* 해당 프로그램은 처음 10진수와 변환해야 할 진수를 입력받고, 변환해야 할 진수에 따라 switch문으로 case가 분리되어 프로그램이 실행되는 구조이다.   
  각 case에는 최종적으로 변환될 진수의 자릿수를 구하는 부분과, 해당 진수 n(2,8, 16)으로 나누어 주면서 그 나머지를 이용하여 변환된 값을 구하는 부분이 있다. 마지막으로 변환된 진수를 출력하는 구문으로 각 case가 구성되어 있다.
* 다음은 선언한 변수에 대한 설명이다.
  + - int num : 변환해야 할 진수
    - int deci : 변환할 10진수
    - char chnum[33] = {} 변환된 수를 출력하기 위해 저장해 놓을 문자열 : deci가 int형으로 입력받으므로, 2, 8, 16진수에서 int형의 최대값(2147483647, 2^31-1)을 충분히 저장할 수 있을만한 크기의 배열(33)을 선언하였다.
    - int dgt : 변환된 진수의 자릿수를 계산하기 위해 선언한 변수. 0으로 초기화(initialize)해둔다. 진수의 자릿수는 해당 진수의 n제곱을 의미하기 때문에(ex. 312에서 3은 3 곱하기 (10의 제곱)을 의미한다) 이후에 자릿수 계산에서 해당 진수를 나누어질때까지 나누면서 나눠질때마다 dgt에 1씩 더하여 최종적으로 변환되는 진수의 자릿수를 구한다)
* 다음은 전체적인 프로그램의 실행과정에 대한 설명이다.

변환할 10진수와 변환시킬 n진수를 입력받는다. 이후 입력한 n에 따라 switch 구문이 실행되어 각 case로 들어가게 된다.

<진법 변환 알고리즘 설명>  
가장 처음 진행하는 것은 최종적으로 변환된 n진수의 자릿수를 구하는 것이다. 10진수를 n의 1제곱부터 제곱, 3제곱..으로 나누어, 몫이 0이 될때까지 나누는 횟수를 구한다. (10진수가 n의 m제곱으로 나누어진다는 것은, 그 수를 n진수로 표현했을 때 (m+1)번째자릿수가 존재한다는 뜻이다. 예를 들어 10은 2의 3제곱인 8로 나누어지기 때문에, 2진수로 표현하면 1001, 즉 4번째 자릿수가 존재한다. 하지만 2의 4제곱인 16으로는 나누어지지 않기 때문에, (4+1)인 5번째 자리는 존재하지 않는다. )

|  |
| --- |
| int decigt = deci;  while (decigt > 0) {  decigt = decigt / 2;  dgt++;  }  dgt--; |

위 코드는 앞서 설명한 자릿수를 구하는 2진수 버전 코드이다. 입력받은 10진수 값으로 initialization을 한 decigt를 이용한다. 그다음 while문이 실행된다. decigt > 0, 즉 10진수를 2로 나누었을 때의 몫이 0에 도달하기 전까지 decigt를 2로 계속 나누어 준다. 그럼과 동시에 main 함수 초반에 0으로 initialization한 변수 dgt에 1을 더해준다. 이 dgt는 decigt가 2로 나누어질때마다 하나씩 늘어난다. 더 이상 2로 나누어떨어지지 않게 되면, 그때의 dgt값이 바로 10진수를 변환한 2진수의 전체 자릿수를 의미한다.

<예외처리>  
만약 0을 2,8,16진수로 변환하고자 한다면, 자릿수를 계산하는 코드에서 (decigt > 0)을 만족하지 못하므로 while문을 실행할 수 없다. 그래서 dgt = 0으로 남게 되고, 바로 아래 “dgt—“에 의해서 dgt값은 -1이 된다. 이렇게 되면 출력을 위한 for문을 실행하지 못하므로 아무것도 출력되지 않게 된다. 0과 같은 다른 한자리 숫자들과 같이 dgt값이 0이 될 수 있도록 예외처리를 진행한다.

|  |
| --- |
| dgt--;  if (deci == 0) {  dgt = 0;  } |

이렇게 되면 while문을 돌지 않고 dgt—로 dgt값이 -1이 되어도 0이기 때문에 dgt = 0으로 다시 intialization된다.

<진수 계산>

2진수 계산

|  |
| --- |
| for (int i = dgt; i >= 0; i--) {  if (deci % 2 == 0) {  chnum[i] = 48;  }  else {  chnum[i] = 49;  }  deci = deci / 2;  } |

위 코드에서 48, 49는 각각 ASCII 코드 상에서 문자 ‘0’과 ‘1’을 의미한다. chnum의 data type이 char형이기 때문에, 배열에 숫자를 기입하려면 ASCII 코드를 이용해야 한다. 즉 위 코드의 의미는, 첫째자리부터 시작하여 마지막 자리까지 2씩 나눠주면서 나머지가 있다면 해당 번째 자리에 들어갈 숫자는 0이고, 나머지가 존재한다면 1이 되는 것이다. 그렇게 마지막 자리까지 숫자 기입을 완료한다.

8진수 계산

|  |
| --- |
| for (int i = dgt; i >= 0; i--) {  if (deci % 8 == 0) {  chnum[i] = '0';  }  else {  int k;  k = deci % 8;  chnum[i] += 48 + k;  }  deci = deci / 8;  } |

1번째 자리부터 시작하여 마지막 자리에 도달할 때 까지 8씩 나눠주면서 나머지가 있다면 해당 번째 자리에 들어갈 숫자는 0이고, 나머지가 존재한다면 8로 나눈 나머지로 초기화시킨 변수 k를 선언하여 chnum[i]에 48+k가 기입될 수 있도록 했다. 이때 48은 ASCII코드에 의해 0이고, 48+k는 숫자 k의 ASCII코드 값이다. 이후 deci를 8로 나눠주어 다음자리의 계산에 진입한다. 그렇게 마지막 자리까지 숫자 기입을 완료한다.

16진수 계산

|  |
| --- |
| for (int i = dgt; i >= 0; i--) {  if (deci % 16 == 0){  chnum[i] = '0';  }  else if(deci % 16 >=10){  int k;  k = deci % 16;  chnum[i] += 55 + k;  }  else {  int k;  k = deci % 16;  chnum[i] += 48 + k;  }  deci = deci / 16;  } |

16진수의 경우 두가지 케이스를 나눠서 생각한다. 나머지가 0~9까지인 경우와 10~15인 경우를 의미한다. 자릿수를 계산하는 과정은 2, 8진수와 같다. 이때 만약 나머지가 0~9인 경우, 48 + 나머지를 통해 ASCII코드 값을 결정하여 array에 기입한다. 만약 10~15인 경우 10, 11, … 15가 각각 A, B, … F에 해당하므로 A의 ASCII코드 값이 65인 것을 이용해 55 + 나머지가 array에 기입할 ASCII 코드값이 되는 것으로 설정하였다.

<출력>

|  |
| --- |
| for (int i = 0; i <= dgt; i++) {  cout << chnum[i];  } |

위 코드는 마지막 과정인 Array output을 하는 과정이다. chnum[0]부터 마지막 chnum[dgt]까지 출력한다. 그럼 n진수로 변환된 수가 출력되게 된다.

1. **토론 및 개선**
   * + 배열을 입출력하는 소스코드를 작성하면서, C++의 cin과 cout이 C언어의 scanf, printf보다 어떠한 점에서 실용적이고 유용한지 알게 되었다.
     + 앞서 예외처리를 진행하기 위해 2진수, 8진수, 16진수 각각의 case마다 dgt=0으로 초기화하는 구문을 작성했는데, 이는 알고리즘적으로 올바르게 설명이 될 수 있는 코드를 작성하기 위함이었다. 좀 더 코드의 길이를 줄이고 알아보기 쉽게 하는 방법으로는 switch 구문을 실행하기 전, 입력받은 deci가 0이라면 바로 0을 출력하는 예외처리 구문을 만들어주면 된다.
     + int type인 deci를 변환했을 때의 변환된 값의 수열을 충분히 저장할 수 있을 만큼 chnum의 길이를 33으로 설정했다. 하지만 작은 값을 변환시킬 때는 Array의 많은 부분이 사용되지 않아 메모리 사용을 낭비하게 된다. 따라서 동적할당을 사용하면 원하는 만큼만 배열을 생성하여 사용할 수 있기 때문에 메모리 낭비를 줄일 수 있을 것으로 기대된다.
2. **참고 문헌**