

# 2023 Spring OOP Assignment Report

과제 번호 : prob1

학번 : 20190445

이름 : 허수범

Povis ID : sbh408

## 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.

## 문제 1번

### 1. 프로그램 개요

본 프로그램은 두 개의 정수를 입력받아 첫 번째 정수(10진수)를 두 번째 정수(2 혹은 8 혹은 16)의 진수로 변환하는 프로그램이다.

프로그램을 실행하면 입력창이 뜨고, 여기에 조건에 맞는 두 개의 정수를 공백을 사이에 두고 입력하고 엔터키를 누른다면, 10진수를 입력한 정수의 진수로 변환한 결과가 콘솔에 출력된다.

masOS를 이용하였기 때문에 프로그램 디렉토리에는 main.cpp와 Makefile이 있다. main.cpp는 본 프로그램의 소스코드이다. Makefile은 Makefile에 작성한 명령어들을 이용하여 main.cpp를 컴파일, 테스트 컴파일, 실행시키거나 디렉토리를 정리할 수 있게 하였다. make clean으로 실행파일을 지울 수 있게 하였고, make testComp는 gcc의 컴파일 옵션을 조정하여 warning을 더 강한 조건으로 띄울 수 있게 하여 디버깅하는데에 도움을 줄 수 있도록 하였다. make all은 main.cpp가 있다면 이를 gcc로 컴파일하여 실행파일을 만들게 하고, make run은 실행파일이 있다면 프로그램을 실행하도록 하였다.

### 2. 프로그램의 구조 및 알고리즘

본 프로그램에서는 main함수를 제외한 세 개의 함수를 선언하여 사용하였다.

첫 번째 함수 `conToBinOrOct`는 10진수를 2진수 혹은 8진수로 변환하는 함수로, 받은 10진수, 밑에 설명할 함수 `binNum`를 통해 계산해낸 반환할 `string`의 길이 `int bits`, `base`(입력이 몇 진수인지 저장된 변수)를 정수로 입력 받는다. 계산을 통해 구한 2진수 혹은 8진수를 `string`타입으로 반환한다.

함수의 동작은 다음과 같다. 먼저 동적 할당을 이용해 `binOct`라는 `bits`만큼의 원소를 가지는 정수형 배열을 선언하였다. 이후 `for loop`를 이용해 각 자릿수를 계산하여 `binOct`에 2진수 혹은 8진수 변환 결과를 저장하였다. 그 다음에는 `binOctResult`라는 `string`을 선언하고 초기화하여, 다시 `for loop`와 `to_string` 함수를 이용해 `binOct`를 `string`으로 변환하여 `binOctResult`에 저장하였다. 이후 동적할당된 메모리를 해제해 주었고, `binOctResult`를 반환하였다.

두 번째 함수 `conToHex`는 10진수를 16진수로 변환하는 함수로, 받은 10진수와 반환할 `string`의 길이 `bits`를 정수로 입력 받고, 16진수를 `string` 타입으로 반환한다. 16진수만 구별해준 것은 16진수 `string`에는 문자가 들어갈 수 있기 때문이다.

함수의 동작은 다음과 같다. 먼저 위의 `conToBinOrOct`와 같이 `hexa`라는 길이 `bits`짜리 배열을 동적할당 해주었고, `for loop`으로 각 자릿수를 계산하여 `hexa`에 변환 결과를 저장해 주었다. 이후 역시 `hexaResult`라는 `string`을 선언하는 동시에 초기화해주었다. 여기서 `conToBinOrOct`와 `conToHex`의 차이가 생기는데, 16진수의 각 자리에는 문자도 들어가기 때문에, `for loop`내에서 `if else`문을 이용하여 `hexa`의 원소가 0부터 9일때와 10부터 15일때를 구분하여 `hexaResult`에 `hexa`의 원소들을 넣어 주었다. 이후 동적할당된 메모리를 해제하고, `hexaResult`를 반환하였다.

세 번째 함수 `bitNum`은 10진수와 `base`를 받으면 해당 `base`진수의 길이가 어떻게 되는지 계산하는 함수이다. 입력된 10진수와 `base`를 입력으로 하며 길이가 정수 타입으로 반환된다.

함수의 동작은 다음과 같다. 먼저 `int num`을 선언하여 입력받은 첫 번째 정수로 초기화해 주고, 자리수를 `count`해주기 위해 `int count`를 선언하고 0으로 초기화해주었다. 이후 `if else`문을 이용하여 입력받은 `base`에 따라 `for loop`를 이용하여 자리수를 세주었다. 이후 `count`를 반환해주도록 하였다.

메인함수에서는 두 개의 정수를 각각 `int deciNum`과 `base`에 입력받고, `bitNum`을 호출하여 `int bits`를 `string`의 길이로 초기화한다. 이후 `if else`문을 사용하여 `input`이 요구사항(두 개의 정수)과 다르거나(이는 `cin.peek()`를 이용해 판별했다), 입력받은 첫 번째 정수가 음수거나, `base`가 2 혹은 8 혹은 16이 아니라면 에러 문구를 출력하고, 요구사항에 부합하는 입력이 들어왔을 시, 각 `base`에 해당하는 함수를 호출하여 `result`를 각 함수의 반환값

으로 초기화하도록 하였다. 이후 result를 출력하도록 하였다.

### 3. 토론 및 개선

Pa1에서는 최대한 cpp언어와 익숙해지려고 노력했다. C에는 없던 string 라이브러리와 그 멤버함수들, new delete를 이용한 동적 할당, cin과 cout의 사용과 그 멤버함수들의 사용 등에 집중했다. 또한, 간단히 결과를 출력하는 것이 아니라, 최대한 예외처리를 많이 해주도록 노력하였다. 고려하지 않아도 되었던 다양한 입력들을 하였을 때 예외처리를 하여 에러문이 출력되도록 하였다(cin.peek()를 활용한 부분이 그 예시). 그 과정에서 cpp 스탠다드 라이브러리의 유용함을 깨달았고, 최대한 많은 입력의 경우의 수를 고려하여 예외처리를 하려면 다양한 경험과 지식이 있어야 함을 깨달았다. 코드의 가독성이나 코드의 단순함, 실행 속도의 향상을 위해서라면 더욱 간단히 프로그램을 작성할 수 있었지만(메인함수만을 사용하거나 공지에 의하면 고려할 필요가 없는 입력에 대한 예외처리를 하지 않는 방법으로), 이번 pa1의 목적은 cpp언어와 친숙해지는 것이라 생각하여 cpp의 다양한 기능들을 사용해보려 노력했다.

또한 이때까지 windows 환경에서 vs로 코딩을 해오다 처음 리눅스 환경에서 vim과 gcc만을 이용하며 개발을 해보았는데, 처음에는 굉장히 힘들고, 시간이 오래 걸린다 느껴졌지만, 다양한 옵션들을 추가해보고 세팅도 해보며 익숙해져 본 결과 고성능 IDE를 사용할 때에 비해 프로그래밍에 대한 이해가 깊어진 것을 느낄 수 있었다.

본 프로그램에 추가할 수 있는 기능은 음수 입력을 추가하는 것, 더욱 다양한 n진수를 구현하는 것, 소수점 이후의 부분도 구현하는 것 등이 있을 것이다. 음수 입력의 경우 먼저 절대값을 씌운 후 계산한 이후 부호를 추가해주면 될 것이고, 다양한 n진수를 입력하는 것의 위의 conTo함수들의 구조를 변경 가능하게 만든 후 하나로 합쳐 다양한 n에 대해 계산이 가능하도록 할 수 있을 것이다. 소수점의 경우 int를 float으로 바꾼 후 간단한 수정을 이용해 바꿀 수 있을 것이다.

### 4. 참고 문헌

<https://blockdmask.tistory.com/334> (to\_string 함수에 대한 자료)

<https://modoocode.com/194> (cin.peek()에 대한 자료)

<https://stackoverflow.com/questions/41020417/cin-peek-for-input-validation>

(cin.peek()에 대한 자료)

