

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 프로그래밍 기초와 실습

과 제 명 : 실습 5: Enumeration Type

담당교수 : 민 형 복 교수

학 과 : 자연과학계열

학 년 : 1학년

학 번 : 2014310407

이 름 : 이준혁

제 출 일 : 2014년 11월 2일

1, Introduction

이번 실습에서는 Enumeration type에 대해 공부한다. Enumerate가 '열거하다'라는 뜻을 가지고 있다는 점에서 알 수 있듯이, Enumeration type이란 그 type의 상수들을 열거하여 선언하는 type을 의미한다. 즉, 사용자가 스스로 자신이 사용할 상수들을 열거하여 정의하는 type으로, user-defined type에 해당한다.

사실 이 Enumeration type은 프로그램이 동작하는 데에는 큰 영향을 주지 못한다. 즉, Enumeration type없이도 충분히 프로그램을 짤 수 있다. 그러나, 자신이 사용할 상수들의 이름을 스스로 지정해서 프로그램을 짤다면, 가독성이 훨씬 좋아질 것이다. Al Kelly, Ira Pohl이 집필한 C by Dissection 4th edition, 242쪽에 "In general, one should use enumerator to aid program clarity"이라 되어 있는 것도 이를 뒷받침한다. 또, 스스로 지정한 상수를 이용하기 때문에 프로그램을 짤 때 훨씬 편리하게 짤 수도 있다.

심지어, typedef라는 identifier를 이용해 자신이 사용할 type의 이름을 지정해 줄 수도 있다. 이것 역시 가독성을 높이는데 굉장히 도움을 준다.

이번 실습에서 만들 프로그램은 컴퓨터와 사용자가 가위바위보를 하는 프로그램이다. 이 과정에서, 가위, 바위, 보 등을 Enumeration type을 이용해 프로그램을 짤다면, 프로그램이 읽기 쉬워지고, 만들기도 쉬워질 것이다.

2. Problem Statement

(A) Describe what the problem is

컴퓨터와 가위 바위 보를 하는 프로그램을 작성하는 것이 주 목적이다.

사용자에게 가위, 바위, 보 중의 하나를 입력하게 하고, 컴퓨터에서 가위, 바위, 보 중 하나를 임의로 생성해 내게 한다. 그래서 누가 이겼는지, 혹은 비겼는지를 비교한다. 그래서 사용자가 무엇을 골랐는지, 컴퓨터가 무엇을 골랐는지, 그리고 누가 이겼는지를 출력하면 된다.

이 때 사용자는 r, s, p 중 하나를 입력함으로 인해 rock, scissors, paper을 주도록 하고, 컴퓨터는 rand함수를 이용해 가위, 바위, 보 중 하나를 내게 한다.

만약 사용자가 q(quit)를 준다면 프로그램을 끝내지만, q를 주기 전까지는 프로그램을 계속 반복해서 실행시킨다. 만약 p, q, r, s 이외의 문자를 주게 된다면, 오류 메시지를 주고 재입력을 요구해야 한다.

이 때 사용되는 가위, 바위, 보 등은 Enumeration type을 이용해야 한다.

(B) Describe how you solve the problem.

처음에 어떻게 풀어야 할지 너무 막막해서, 민형복 교수님의 program template p5.c를 참고하였다. 그래서 우선 가위, 바위, 보, 탈출을 의미하는 RSP라는 이름의 Enumeration type과, 누가 컴퓨터가 이겼는지, 사용자가 이겼는지, 혹은 비겼는지를 나타내는 WHO_WIN이라는 이름의 Enumeration type을 선언하기로 했다.

그리고 어떤 function을 쓸지 잘 모르겠어서 function prototype은 프로그램을 작성하면서 생각하기로 하고, main 함수를 어떻게 작성할까 생각해 보았다.

우선 반복적으로 수행해야 하므로, while문을 이용하기로 했다. 그리고 사용자에게 값을 받아들이려고 했는데, 이렇게 하면 프로그램이 조금 복잡해 보일 것 같았다. 왜냐하면 거기에 q인 경우와 p, q, r, s가 아닌 경우를 나타내면 아무래도 쓸게 많아질 것 같았기 때문이다. 그래서, 교수님의 template에서 착안한 아이디어로 함수를 이용해 표현하기로 했다. 함수 내에서 사용자에게 입력을 요구하고, 받아들이어서 잘못된 입력이면 오류 메시지를 주고, 제대로 된 입력을 준 경우에는, 사용자가 입력한 것에 따라 가위, 바위, 보 값 혹은 탈출 값을 반환하는 함수를 만들기로 했다.

컴퓨터가 내놓는 값에 대해서도, 마찬가지로 함수를 이용하기로 했다. 다만 문제에서도 주어진 것과 같이, 임의의 값을 주기 위해 random함수를 이용하기로 했다. 다만 random함수가 내놓는 값은 정수type의 값이고, 컴퓨터가 나타내는 값은 가위, 바위, 보 중의 하나로 RSP라는 type의 값이기 때문에, type casting을

잘 이용해야 할 것 같았다.

이렇게까지 하면 사용자가 준 값과, 컴퓨터가 내놓은 값 두 개가 있을 테니 이 두 개의 값을 비교해 그에 따라 사용자 승리, 컴퓨터 승리, 비김을 내놓는 함수를 작성하기로 했다. 쉬울 줄 알았는데, 생각보다 만들기가 어려웠다. 그래서 한참 고민한 끝에, 사용자의 값을 정수로 type casting한 것에서 컴퓨터의 값을 정수로 type casting한 것을 빼서, 그 값에 따라 누가 이겼는지 판단할 수 있다는 것을 알게 되었다. 따라서, 이를 이용하기로 했다. (Implementation을 참고해 주세요)

이렇게 누가 이겼는지 까지 파악하는 프로그램을 만드는 방법은 완벽히 구상했으나, 출력이 문제였다. 사용자가 무슨 값을 골랐는지, 컴퓨터가 무슨 값을 골랐는지 출력하는 것도 생각만큼 쉽게 구상되지 않았다. 하지만 방법을 결국 찾아냈는데, 바로 function call을 통해 값이 결정되면, 바로 밑에 if - else문을 이용해 그 값에 따라 한 구절씩 출력하는 방법으로 하기로 했다. 예를 들어서, 사용자의 값이 paper로 결정되면, You choose paper까지만 출력하는 방식으로 했다. (프로그램 코드와 Implementation을 참고해 주세요)

3. Implementation

가장 먼저 할 일은 헤더 파일 삽입이었는데, 여태까지는 <stdio.h>만 이용했다면, 이번 프로그램에서는 random함수를 이용하기 위해 <stdlib.h>까지 이용했다.

그리고 프로그램 진행에 필요한 두 개의 Enumeration type을 설정했다. 첫 번째로는 프로그램을 진행하는데 꼭 필요한 rock, scissors, paper 그리고 quit 이 네 개의 상수를 가지는 type을 typedef를 이용해 RSP로 선언했다. RSP는 rock, scissors, paper의 앞 글자 들에서 따왔다. 또, 누가 이겼는지를 설명하는 user_win, computer_win, tie 이 세개의 상수를 가지는 type을 WHO_WIN으로 선언했다. 각각의 상수 이름과 type이름은 민형복 교수님의 program template를 참고했다.

그리고 문제 해결 방법을 구상할 때 생각했던 함수 3개를 function prototype 자리에 작성했다. 첫 번째 함수 사용자의 가위, 바위, 보, 혹은 탈출(quit)값을 반환하는 함수였다. 이 때 이 함수의 필요한 입력들은 함수 안의 scanf를 이용해서 받으니, 함수 자체의 입력은 필요 없다고 생각해 입력이 없는 함수로 됐다. 이름은 사용자의 가위, 바위, 보라는 뜻에서 USER_RSP로 됐다. 두 번째 함수는 random 함수를 이용해 컴퓨터의 가위, 바위, 보 값을 반환하는 함수였다. 이 함수는 random 함수를 통해 값을 만들어 내므로, 입력이 있을 리가 없었다. 함수의 이름은 컴퓨터의 가위, 바위, 보라는 뜻에서 COMPUTER_RSP로 됐다. 마지막으로 위 함수 두 개 에서 반환된 값들로 입력을 받아서 누가 이겼는지를 결정해 WHO_WIN type으로 반환하는 decideWhoWins라는 이름의 함수를 선언했다. 이 이름 역시 교수님의 template의 것을 참고했다.

이제 메인함수 내부를 구성해 보았다. 먼저 각 function call된 값을 저장할 변수들을 선언했다. USER_RSP가 반환한 값을 저장하기 위한 사용자의 값이라는 뜻을 가진 RSP type의 변수 USER_VALUE를 선언했다. 그 다음에

COMPUTER_RSP가 반환한 값을 저장하기 위한 컴퓨터의 값이라는 뜻을 가진 RSP type의 변수 COMPUTER_VALUE를 선언했다. 마지막으로 decideWhoWins의 반환 값을 저장하기 위해 누가 이겼는지 결과를 나타낸다는 뜻으로 WHO_WIN type의 변수 result를 선언했다.

그리고 이 프로그램은 계속 반복되어야 하므로, main함수의 내용들을 모두 while문 안에 집어넣었다. 그리고 while(1)로 조건식이 항상 참이 되게 하였다. 왜냐하면 사용자가 q를 쳤을 때 만 프로그램을 break문을 이용해 탈출하기 위해서이다. 처음에는 while 뒤의 조건식을 적절히 변형해볼까 생각도 해봤지만, 오히려 더 복잡해지고 가독성도 떨어져서 하지 않기로 했다.

첫 번째로 사용자가 주는 값을 구하기 위해 다음과 같이 function call을 했다.

```
USER_VALUE = USER_RSP();
```

그리고 이에 상응하는 USER_RSP라는 함수를 작성했다. (main 함수 밖에)

이 함수는 입력은 없지만, 사용자에게 입력을 요구하고, 또 입력을 받아들여야 했다. 그래서 일단 사용자에게 입력을 받아들이기 위해 사용자가 준다는 의미로 usergive라는 문자형 변수를 선언했다. 그리고 이 변수를 RSP type으로 저장하기 위해 USER_VALUE라는, 사용자의 값이라는 뜻의 RSP type의 변수를 선언했다. 이 변수가 main함수의 USER_VALUE라는 변수와 다르다는 사실을 알고 있었기에 사용이 가능했다.

그리고 사용자가 가위, 바위, 보 혹은 탈출 중 어떤 걸 낼지에 대해 입력을 요구해야 했으므로, 이 부분은 늘 하던 것과 같이 printf함수와 scanf함수를 이용해 작성했다. 이 때 사용자가 입력한 enter키를 무시하고 입력을 받아들이기 위해, " %c"로 작성하는 것도 잊지 않았다. 그 후 scanf를 이용해 usergive로 받아들이는 값이 무엇인지 파악해야 했다. 만약 이 값이 'p', 'q', 'r', 's'중 하나가 아니라면 재입력을 요구해야 했으므로, while문을 이용해 제대로 된 입력을 줄 때까지 오류 메시지를 주고 다시 입력하기를 요구했다. 이렇게 while문이 끝나면, switch문을 이용해 사용자가 준 p, q, r, s에 따라 USER_VALUE에 그 값을 할당했다. 그리고 USER_VALUE를 반환하기로 했다.

이렇게 작성하면 main함수의 USER_VALUE라는 변수에 ROCK, SCISSORS, PAPER, QUIT중 하나가 들어가 있을 것이다. 만약 이 변수가 QUIT이면 위에서 언급한 것처럼 break문을 이용해 탈출하기로 했다. 나머지 경우는 사용자가 무엇을 골랐는지 안내해 주는 문구를 출력하기로 했다. 다음과 같이 작성했다.

```
if (USER_VALUE == QUIT)

    break;

else if (USER_VALUE == ROCK)

    printf("you choose rock ");

else if (USER_VALUE == SCISSORS)

    printf("you choose scissors ");

else if if (USER_VALUE == PAPER)

    printf("you choose paper ");
```


위에서도 언급했지만, 이렇게 한번 값이 할당될 때 마다 한 구절씩 작성하기로 했다.

두 번째로 컴퓨터가 내놓는 값을 구하기 위해 역시 비슷한 구성으로 작성했다. 다음과 같이 function call을 했다.

```
COMPUTER_VALUE = COMPUTER_RSP();
```

그리고 역시 이에 상응하는 COMPUTER_RSP라는 함수를 작성했다. (main 함수 밖에)

이 함수는 문제에서도 제시되었지만, random함수를 이용해 임의로 값을 정해야 했다. 그래서 일단 random함수의 값을 저장할, random함수의 이름에서 따온 정수형 변수를 random을 선언했고, 이 변수를 RSP type으로 저장할, 컴퓨터의 값이라는 의미의 COMPUTER_VALUE를 선언했다. 그런데 random함수를 통해 가위, 바위, 보 중의 하나의 값을 얻고 싶었기 때문에, random함수가 내놓는 값을 3개로 한정해야 되었다. 따라서, 다음과 같이 변수 random에 값을 할당했다.

```
random = (rand()) % 3;
```

그러면 random값은 0, 1, 2중 하나가 될 것이다. 그런데 RSP의 Enumeration type을 선언할 때, 다음과 같이 선언했다.

```
typedef enum {ROCK, SCISSORS, PAPER, QUIT} RSP;
```

이 때 ROCK은 0, SCISSOR은 1, PAPER은 2의 값을 가지게 된다. 즉,

COMPUTER_VALUE에 random의 결과로 나온 값을 할당하면, ROCK, SCISSORS, PAPER값 중 하나를 가지게 될 것이었다. 다만 여기서 혹시 type이 변경되면서

오류가 생길 수 있으므로, 그런 오류를 방지하기 위해 (RSP)라는 type casting을 덧붙였다. 그렇게 할당한 후에, COMPUTER_VALUE를 반환 하기로 했다.

이렇게 작성하면 main함수의 COMPUTER_VALUE라는 변수에 ROCK, SCISSORS, PAPER 중 하나가 들어가 있을 것이다. 위에서 한 것과 마찬가지로 컴퓨터가 어떤 것을 내놓았는지 안내해 주는 문구를 출력하기로 했다. 다음과 같이 작성했다.

```
if (COMPUTER_VALUE == ROCK)

    printf("and I choose rock. ");

else if (COMPUTER_VALUE == SCISSORS)

    printf("and I choose scissors. ");

else if (COMPUTER_VALUE == PAPER)

    printf("and I choose paper. ");
```

이제 사용자가 어떤 값을 줬는지, 컴퓨터가 어떤 값을 내놓았는지가 다 결정이 되었으므로, 위에서 얘기한 대로 decideWhoWins라는 함수를 통해 누가 이겼는지를 결정하기로 했다. 다음과 같이 function call을 했다.

```
result = decideWhoWins(USER_VALUE, COMPUTER_VALUE);
```

result의 type과 decideWhoWins가 반환하는 type이 모두 WHO_WIN type이므로, result는 USER_WIN, COMPUTER_WIN, TIE셋 중의 하나를 가질 것이다. 이제 USER_VALUE와 COMPUTER_VALUE의 값에 따라서 decideWhoWins가 USER_WIN,

COMPUTER_WIN, TIE중 하나를 알맞게 반환하도록 main 함수 밖에 function definition을 작성했다.

문제 해결에서 구상한 것처럼, 사용자의 값을 정수로 type casting한 것에서 컴퓨터의 값을 정수로 type casting한 것을 뺀 값을 가지는 변수를 선언하기로 했다. 이 값이 사용자가 이겼는지, 컴퓨터가 이겼는지, 비겼는지를 결정하기 때문에, decision number라고 이름을 붙였다. 즉, 다음과 같이 선언했다.

```
int decision_number;
```

```
decision_number = ((int)USER_VALUE - (int)COMPUTER_VALUE);
```

이제 사용자와 컴퓨터의 값에 따라 decision number와 누가 이기는지를 표로 정리했다.

사용자 컴퓨터	ROCK	SCISSOR	PAPER
ROCK	Decision number : 0, 무승부	Decision number : 1, 컴퓨터 승	Decision number : 2, 사용자 승
SCISSOR	Decision number : -1, 사용자 승	Decision number : 0, 무승부	Decision number : 1, 컴퓨터 승
PAPER	Decision number : -2, 컴퓨터 승	Decision number : -1, 사용자 승	Decision number : 0, 무승부

[표 1] 사용자와 컴퓨터의 값에 따른 decision number와 승패 여부

즉, decision number가 0일 때는 무승부, 1 또는 -2일 때는 컴퓨터 승, -1 또는 2일 때는 사용자 승이다. 이에 기반해서, 다음과 같이 프로그램을 짰다.

```
if (decision_number == 0)

    return TIE;

else if ((decision_number == -1) || (decision_number == 2))

    return USER_WIN;

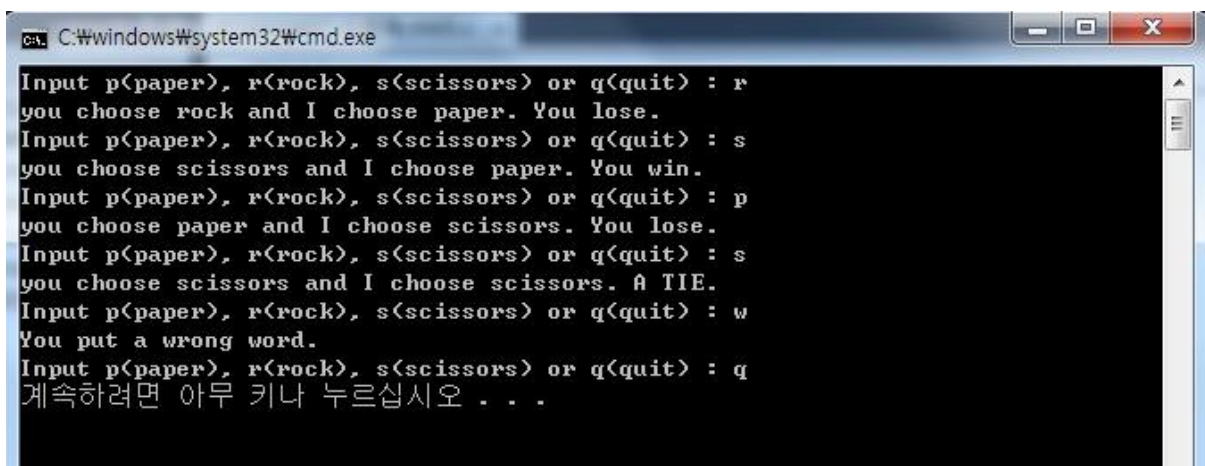
else if ((decision_number == 1) || (decision_number == -2))

    return COMPUTER_WIN;
```

이렇게 하면 main함수의 result라는 변수에 TIE, USER_WIN, COMPUTER_WIN중 하나가 들어가 있을 것이다. 이제 이 결과에 따라, 위에서 했던 것처럼 알맞은 출력을 하기로 했다.

이렇게 프로그램을 작성했고, 실행해 보았다.

4. Result



```
C:\windows\system32\cmd.exe
Input p(paper), r(rock), s(scissors) or q(quit) : r
you choose rock and I choose paper. You lose.
Input p(paper), r(rock), s(scissors) or q(quit) : s
you choose scissors and I choose paper. You win.
Input p(paper), r(rock), s(scissors) or q(quit) : p
you choose paper and I choose scissors. You lose.
Input p(paper), r(rock), s(scissors) or q(quit) : s
you choose scissors and I choose scissors. A TIE.
Input p(paper), r(rock), s(scissors) or q(quit) : w
You put a wrong word.
Input p(paper), r(rock), s(scissors) or q(quit) : q
계속하려면 아무 키나 누르십시오 . . .
```

결과에서는 다음 세 가지 사항을 점검해 보려 했다.

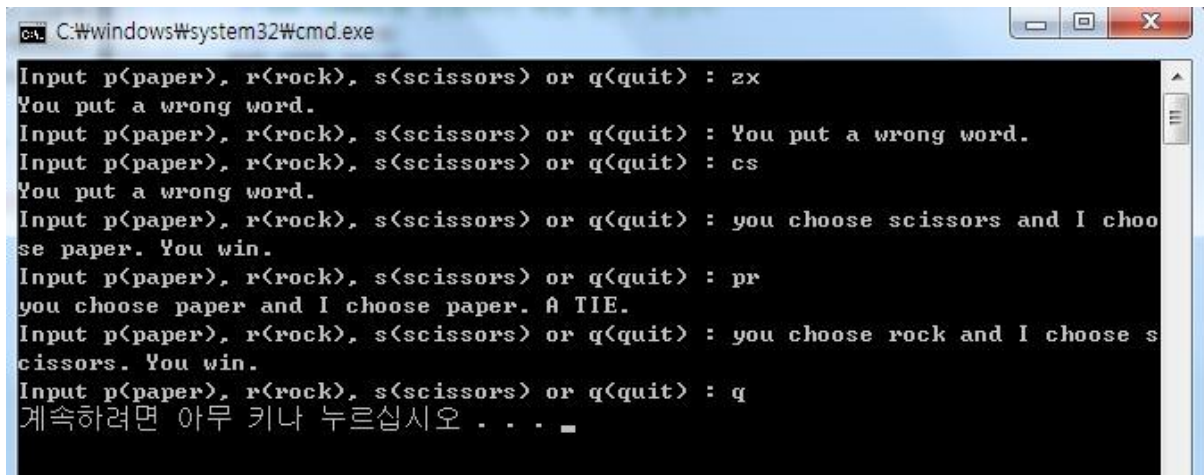
1. 사용자가 준 값과 컴퓨터가 내놓은 값에 따라 적절한 문구를 출력하고, 누가 이겼는지 혹은 비겼는지를 제대로 출력하는가
2. p, q, r, s가 아닌 값을 줬을 때 재입력을 요구하는가
3. q를 줬을 때 프로그램을 끝내는가

첫 번째 것을 점검하기 위해, r, s, p, s로 줘 보았다. "you choose ...(사용자가 준 값)"으로 나오면서, 사용자가 준 값에 맞춰 출력되는 것을 볼 수 있었다. 또, 매번 입력할 때마다 컴퓨터가 내놓는 값이 달라지는 것도 확인했다. 그리고 사용자가 묵, 컴퓨터가 보 일 때 사용자 패, 사용자 가위, 컴퓨터 보 일 때 사용자 승으로 출력하는 등 사용자와 컴퓨터의 가위, 바위, 보에 따라 승패 여부를 정확하게 출력했다.

두 번째 것을 점검하기 위해 w를 입력해 보았는데, you put a wrong word라는 메시지를 주면서 다시 입력할 것을 요구했다.

세 번째 것을 점검하기 위해 q를 줬는데, 프로그램이 끝났다. 예상대로였다.

그러나 아쉬운 부분은 없지 않았다. 바로 두 개 이상의 문자를 연속해서 입력하면, 다음과 같이 출력되었다.



```
C:\windows\system32\cmd.exe
Input p(paper), r(rock), s(scissors) or q(quit) : zx
You put a wrong word.
Input p(paper), r(rock), s(scissors) or q(quit) : You put a wrong word.
Input p(paper), r(rock), s(scissors) or q(quit) : cs
You put a wrong word.
Input p(paper), r(rock), s(scissors) or q(quit) : you choose scissors and I choose paper. You win.
Input p(paper), r(rock), s(scissors) or q(quit) : pr
you choose paper and I choose paper. A TIE.
Input p(paper), r(rock), s(scissors) or q(quit) : you choose rock and I choose scissors. You win.
Input p(paper), r(rock), s(scissors) or q(quit) : q
계속하려면 아무 키나 누르십시오 . . .
```

바로 문자를 하나씩 판단해서, 그 각각의 문자에 대한 출력을 주는 것이었다. 예를 들어, zx를 입력한 경우 오류 메시지를 두 번 주었고, cs와 같이 입력한 경우에는 처음에는 오류 메시지를 주고, 두 번째에는 s를 가위로 인식해서 출력했다. 그리고 pr처럼 두 개 다 한 문자로 인식 가능한 경우에는 결과를 두 번 출력하였다. 이런 문제 때문에 두 개 이상의 문자를 입력할 시에는 오류 메시지를 주고 싶었으나, 별다른 방법이 없어서 포기했다.

5. Conclusion & Evaluation

이번 실습은 2개 이상의 단어를 줄 때, 좀 아쉬운 결과가 났다는 것을 제외하면, 굉장히 만족스러운 실습이었다. 사실 2개 이상의 결과를 찾을 때도 심각한 오류가 생기는 것도 아니고, 원래의 프로그램 가정을 어느 정도 따르기 때문에, 저 문제도 그렇게 큰 문제는 아니다. 하지만 이 문제를 어떻게 해결할 수 있는지 더 생각해 봐야 할 것 같다. 여태까지 배운 것으로 해결할 수 있는지, 그것이 불가능하다면 앞으로 더 공부한 다음에 이 문제를 해결할 수 있는지를 알아봐야

할 것 같다.

또, 이번 실습을 통해서 Enumeration type에 대해 자세히 알 수 있었다. 실습 전에는, Enumeration type에 대해 이해는 하고 있었지만, 뭔가 막연한 감이 없잖아 있었는데, 이번 실습을 통해 완벽히 감을 잡을 수 있게 되었다. 게다가 Enumeration type이 얼마나 가독성을 높이는 지에 대해 몸소 깨달았다. 가독성을 높일 뿐만 아니라, 프로그램 할 때 헛갈릴 일이 적어지니, 훨씬 더 편리하기도 했다.

또 이번 실습에서 style과 comment가 얼마나 중요한지도 알게 된 것 같다. 예전에는 이렇게 긴 프로그램을(사실 이것도 아주 짧은 프로그램이지만, 예전에 만들었던 프로그램에 비해서는 긴 편이니) 만들어 본 적이 없어서, 왜 style이나 comment가 중요한지 미처 몰랐다. 하지만 이렇게 프로그램이 길어지면, style을 맞춰 쓰지 않거나 comment가 없으면 정말 읽기 힘들어 질 것 같아 보였다. 앞으로 더 이런 부분에 신경을 써서, 프로그램의 가독성을 높일 수 있도록 해야겠다.

6. 참고 문헌

[1] 민형복, program template, p5.c.

[2] 프로그래밍 기초와 실습 사이트, <http://class.icc.skku.ac.kr/~min/C/>, 보고서 작성
요령, 2014 년 10 월.

[3] Al Kelly, Ira Pohl, *C by Dissection: The Essentials of C Programming, Fourth Edition*, Pearson, p. 241 ~ 243.

[4] 민형복, practice5.pdf.