

C++프로그래밍및실습

# 전투력측정기

진척 보고서 #1

제출 일자 : 12/03

이름 : 한승한

학번 : 213996

## 프로젝트 목표

### 배경 및 필요성

코딩에 대한 배경지식이 거의 없어서 지금까지 배운 내용들을 기반으로 프로젝트를 진행하려고 했다.  
그러다 보니 단순하면서 실용성 있는 프로젝트를 생각하게 되었다.

운동을 시작하였는데 실제로 근성장이 이루어지고 있는지 궁금하여 확인이 필요하였고,  
이에 근성장이 제대로 이루어지는지 판단하는 프로그램이 필요하여 직접 만들어보기로 하였다.

### 프로젝트 목표

데이터가 주어졌을때 그것을 기반으로 얼마나 근력이 상승했는가에 대한 수치(단위 %)  
를 나타내는 프로그램을 만드는 것이 목표이다.

### 차별점

물론 흔히 일지를 작성하는 프로그램이 존재한다. 하지만 거의 일지만 작성할 수 있는 프로그램이며  
대부분 VER.Pro와 같은 유료화 된 버전을 사용해야만 수치가 나타난다.  
금전적 부분에서 차별이 있다고 생각한다.

## 기능 계획

### STEP 1. 시작 단계

사용자가 접근하기 쉽도록 길잡이가 필요하였고 이에 메뉴의 기능이 필요하다.

세부 기능 1)

선택하는 메뉴창을 띄운다. ( 1. 종합적으로 계산 / 2. 분할적으로 계산 )

선택 1) 바로 **STEP 2)**로 넘어간다.

선택 2) 밑의 세부 기능 2)로 넘어간다.

세부 기능 2)

메뉴창을 출력한다. ( 1. 등 / 2. 가슴 / 3. 어깨 / 4. 하체 / 5. 이두 / 6. 삼두 / 7. 복근 )

사용자가 올바른 번호 입력을 마치면 **STEP 2)**로 넘어간다.

세부 기능 2-1)

추가 기능을 생성한다. ( 입력 : /추가 )

만일 메뉴의 목록 중 자신이 추가하고자 하는 목록이 있는 경우 추가를 할 수 있도록 도와준다.

ex) 전완근을 추가하고자 한다. ( 입력: /추가 -> 입력 : 전완근 )

( 1. 등 / 2. 가슴 / 3. 어깨 / 4. 하체 / 5. 이두 / 6. 삼두 / 7. 복근 / 8. 전완근 )

세부 기능 3)

그 외의 입력일 경우 경고문을 보내고 다시 입력으로 돌아온다.

### STEP 2. 단위 선택 단계

사용자의 입력을 받기 전에 단위를 선택한다. ( 1. 달 / 2. 주 )

요일마다 받는 것은 수치가 굉장히 많아진다.

또한 근성장은 곧바로 일어나지 않기 때문에 큰 단위를 사용하여 입력받기 위함이다.

세부 기능 1)

단위를 입력받는다.

세부 기능 2)

그 외의 입력일 경우 경고문을 보내고 다시 입력으로 돌아온다.

### STEP 3. 사용자 입력 저장

사용자가 입력한 수치를 저장한다.

하지만 사용자가 입력법을 모르기 때문에 안내문을 출력한다.

세부 기능 1)  
안내문 출력

세부 기능 2)  
사용자로부터 입력받기

세부 기능 3)  
만일 숫자가 아닌 다른 값이 입력된다면 경고문을 출력하고 다시 입력으로 돌아온다.

### STEP 4. 사용자 DATA 분석

사용자의 데이터 값을 분석한다. 대단한 분석법을 쓰는 것이 아니고 그냥 상승하는 정도치만 계산한다.

세부 기능 1)  
입력받은 자료의 차이를 백분율로 바꿔서 계산시킬 것 같다.

### STEP 5. 결과

분석값에 대한 결과를 표기해준다.

세부 기능 1) cout  
그냥 cout 기능을 사용해서 출력한다.

세부 기능 2) graph  
색이 채워진 블록과 그렇지 않은 블록을 사용해 그래프를 만들어볼 예정이다.

고려해야 할 점.

- \* 음수값이 발생 할 경우도 대비를 해야 한다. -> 오히려 근손실이 발생했을 경우
- \* 1달을 기준으로 측정하기에 측정 결과값이 매우 미미할 수 있다.
- \* 시험기간이나 과제로 인한 시간적 여유가 없을 경우를 대비해 넉넉한 일정으로 계획을 세운다.
- \* 자료값은 실제 나의 운동값을 사용할 것이므로 운동을 마친 후 일지를 꼭 작성해야 한다.

## 진척사항

### STEP 1. 시작 단계

#### < 함수 설명 1.>

```
Menu::Menu() { // Menu 객체 생성과 동시에 부위를 벡터에 추가
    for (int i = 0; i < 7; i++) {
        devide.push_back(part[i]); // 전역변수로 사용한 배열을 추가
    }
}
```

설명)

객체가 생성될 때 자동으로 호출되는 함수다.  
전역변수로 사용한 배열을 벡터<sup>1)</sup>에 추가하는 구문이다.  
배열<sup>2)</sup>은 총 7개로 이루어져 있으며 다음과 같다.

```
// 전역 변수 : string 부위 배열 : 10/14 string
const string part[] = {"1. 등", "2. 가슴", "3. 어깨", "4. 하체", "5. 이두", "6. 삼두", "7. 복근"};
```

- 사용한 배열 상 -

#### < 함수 설명 2.>

```
void Menu::print_1() { // 세부기능 1) 사용 문구
    enter();
    cout << "성장치를 측정하는 프로그램입니다." << endl;
    cout << "아래의 선택사항 중 하나를 번호로 입력하세요.";
    enter();
    cout << "  1. 종합적으로 계산";
    enter();
    cout << "  2. 분할적으로 계산" << endl;
    enter();
}
```

설명) 말 그대로 프린트 함수다. cout을 통해 사용자에게 안내문을 출력한다

---

1) 10주 vector

2) 6주 string, 11주 const 키워드, 5주 Array

### < 함수 설명 3.>

```
void Menu::print_2() { // 세부기능 2) 사용 문구
    enter();
    cout << "아래의 선택사항 중 하나를 번호를 입력하세요." << endl;
    cout << "추가하고 싶은 목록이 있다면 '/add'를 입력하세요.";
    enter();
    for (auto& e:devide) { // 벡터 주소값을 받아 출력
        cout << "    " << e;
        enter();
    }
}
```

설명) 사용자에게 문구를 출력하는 함수다. 더불어 auto&<sup>3)</sup>를 통해 devide의 주소값을 받아 출력한다.  
사용한 devide는 Menu클래스<sup>4)</sup>에 public으로 선언된 벡터다.

```
vector<string> devide; // string 벡터 생성 : 11/06 Vector
```

- vector : devide 上 -

### < 함수 설명 4.>

```
bool Menu::check_number_ver1() { // 번호 검사하는 함수 : version 1 - 세부기능 1) 사용
    please(); // 안내문 출력

    // int형이 아닌 문자 입력시 오류 방지
    if (!(cin >> number)) { // 입력 : int(X)
        cin.clear(); // 오류 초기화
        cin.ignore(10000, '\n'); // 값 무시
        enter();
        cout << "다시 입력하세요." << endl;
        return true; // true반환 : 다시 반복
    }

    // 숫자 구분
    if(number == 1) { // 입력 : 1
        enter();
        next_page();
        return false; // false반환 : 반복문 탈출
    }
    else if(number == 2) { // 입력 : 2
        enter();
        next_page();
        return false; // false반환 : 반복문 탈출
    }
    else { // 입력 : 1과 2를 제외한 '숫자들'
        enter();
        cout << "다시 입력하세요." << endl;
        return true; // true반환 : 다시 반복
    }
}
```

설명) 번호를 검사하는 함수다.

cin을 통해 입력받은 값이 int형인지 검사하여 오류를 없애준다.

입력받아 저장하는 변수가 int형이기 때문에 필요한 함수이다.

만일 int형이 아닐시 오류를 초기화하고 받은 값을 무시한 뒤 다시 입력받는다.

1 또는 2를 입력할 시 반복문을 탈출하여 다음 스텝으로 넘어간다.

그 외의 숫자들을 입력시 true값을 반환하여 반복문을 계속 순환한다.

---

3) 5주 auto, 4주 for

4) 9주 Class

위 코드에서 사용한 변수와 함수들은 다음과 같다.

```
void Menu::please() { // 안내문
    number = 0; // 매 질문문이 반복될 때마다 number값 초기화
    cout << "번호를 입력하세요. : ";
}
```

\* 번호를 입력하라는 질문이 많이 필요하여 만든 함수다. number의 값을 초기화하는 역할도 추가하였다. number로 계속해서 번호를 입력받기 때문에 초기화가 필요하였다.

```
int number; // 입력받은 번호를 저장할 변수
```

\* Menu Class에 Public으로 선언된 변수다.

```
void Menu::enter() { // 이중 Enter
    cout << endl << endl;
}
```

\* 이중 줄바꿈을 진행하는 함수다.

```
void Menu::next_page() { // NEXT PAGE print
    cout << "┌─────────── < NEXT PAGE > ───────────┐" << endl;
}
```

\* 칸을 나누기 위한 그림을 출력하는 함수다.

### < 함수 설명 5.>

```
bool Menu::check_number_ver2() { // version 2 - 세부기능 2-1) 사용
    if(number >= 1 && number <= devide.size()) { // 벡터 사이즈 안의 번호를 입력시 출력하고 반복문 빠져나오기
        enter();
        cout << devide[number - 1] << "(을/를) 선택하셨습니다.";
        enter();
        return true;
    }
    else { // 벡터 사이즈 밖의 번호 입력시 반복
        enter();
        cout << "다시 입력하세요." << endl;
        return false;
    }
}
```

설명) 번호를 검사하는 함수<sup>5)</sup>다.

입력받은 숫자가 벡터 인덱스면 해당되는 값을 출력하고 반복문을 종료한다. 그 밖의 번호를 입력시 반복문을 순환한다.

---

5) 4주 if, else

### < 함수 설명 6.>

```
bool Menu::check_add() { // 추가 명령어 - 세부기능 2-1) 사용
    if (input == "/add") { // '/add' 명령어 입력시 안내문을 출력하고 부위 입력받기
        enter();
        cout << "한글이 깨지기 때문에 영어로 입력바랍니다." << endl;
        cout << "추가할 항목을 입력하세요 : ";
        cin >> answer;
        count = devide.size() + 1; // 인덱스 오차
        devide.push_back(to_string(count) + ". " + answer); // 다음 인덱스에 추가하기
        enter();
        print_2(); // 추가한 목록 다시 보여주기
        return true;
    }
    return false;
}
```

설명) add라는 입력이 들어왔는지 검사하는 함수며, /add라는 입력이 들어올 시 실행된다.  
안내문을 출력하고 추가할 항목을 입력받아 벡터에 추가<sup>6)</sup>한다.

```
int count; // 사용자가 입력한 부위를 추가할 때 앞에 붙을 숫자 변수
string answer; // 추가할 부위를 저장할 변수
```

\* 사용한 answer, count 변수는 Menu Class에 Public으로 선언된 변수다.

### < 함수 설명 7.>

```
void Menu::check_input() { // input 숫자 처리
    if(is_number(input)) { // 숫자인지 확인
        number = stoi(input); // 숫자로 변환하여 number에 저장
    }
}
```

설명) is\_number()함수<sup>7)</sup>를 통해 입력받은 값이 숫자인지 확인한다.  
만일 숫자면 String인 input을 int형으로 형변환하여 변수에 저장한다.  
(input을 형변환하는 이유는 형변환하여 number(int)변수에 저장시키고 check\_number함수를 사용하기 위함이다.)

```
string input; // 입력받은 명령을 저장할 변수
```

\* 사용한 input 변수는 Menu Class에 Public으로 선언된 변수다.

---

6) 7주 string의 합

7) 11주 Pointer, 9주 정보는닉 - 밑 페이지에서 함수 설명



## is\_number() 함수

```
bool Menu::is_number(const string& str) { // 문자열이 숫자인지 확인하는 함수
    if (str.empty()) // 문자열이 비어있는지 확인 // 4주차 Condition
        return false; // false 반환
    for (char ch : str) { // 문자열 배열 반복 // 5주차 Loop and Array
        if (!isdigit(ch)) { // 현재 문자가 숫자가 아닌지 확인
            return false; // 숫자가 아니면 false 반환
        }
    }
    return true; // 둘 다 해당되지 않으면 true 반환 : 다시 반복
}
```

설명) 대단한 것은 아니고 Menu Class에 Private로 선언된 함수다.

입력받은 string형 input을 매개변수로 넣는다. input의 입력이 없거나 숫자가 아니면 false를 반환한다. true를 반환하면 위 check\_input함수에서 형변환당하여 number에 저장하게 된다.

\* input은 '/add'라는 문자형도 받아야하고 1,2라는 숫자도 받아야 하여 string으로 선정하게 되었다.

## STEP 2. 단위 선택 단계

### < 함수 설명 1.>

```
void Menu::STEP_2() { // 단위 선택 안내문 출력
    enter();
    cout << "주기를 선택하세요." << endl;
    cout << "1달 주기로 분석을 원하시면 1번, 1주 주기로 분석을 원하시면 2번을 선택해주세요.";
    enter();
    cout << "  1. 달";
    enter();
    cout << "  2. 주" << endl;
    enter();
}
```

설명) STEP 2단계에 도달하면 출력되는 안내문이다. cout로 출력한다.

## < 함수 설명 2.>

```
bool Menu::check_number_ver3() { // version 3 - STEP 2 사용
    while(true) {
        please(); // 번호를 입력해 달라는 안내문

        // 위 방법과 똑같은 방법으로 int형이 아닌 문자가 들어올 시 오류를 초기화하고 입력값 무시
        if(!(cin >> number)) {
            cin.clear();
            cin.ignore(10000, '\n');
            enter();
            cout << "다시 입력하세요." << endl;
            return true;
        }
        if(number == 1) { // 입력한 숫자가 1
            // 정보 입력 단계로 넘어가기
            return false;
        }
        else if(number == 2) { // 입력한 숫자가 2
            // 정보 입력 단계로 넘어가기
            return false;
        }
        else { // 그 외의 다른 숫자들 처리
            enter();
            cout << "다시 입력하세요." << endl;
            return true;
        }
    }
}
```

설명) 번호를 검사하는 함수다.

check\_number\_ver1과 동일한 방법으로 동작한다.

number변수에 자료형이 int인지 확인하고 int형이면 반복문을 탈출하고, 그렇지 않으면 반복한다.

그 밑으로는 상황에 맞는 번호를 입력하면 반복문을 탈출하고 다음으로 넘어가며, 그렇지 않을 경우 입력을 계속 반복하게 된다.

### STEP 3. 사용자 입력 저장

```
string path; // 파일 경로를 입력받을 string 변수
```

\* 앞으로 받을 경로를 저장할 변수다. Menu Class에 Public에 존재한다.

#### < 함수 설명 1.>

```
void Menu::input_path() { // 경로 입력받기
    if(cin.peek() == '\n') // 버퍼 안에 줄바꿈 문자가 있다면
        cin.ignore(); // 무시하기
    getline(std::cin, path); // 파일 경로 입력받기
}
```

설명) 경로를 입력받는 함수다.

우선 버퍼 안에 줄바꿈문자가 존재한다면 그것을 무시하고 경로를 입력받는다. 줄바꿈 무시 코드가 없다면 안내 문구가 꼬이게 되어 놓게 되었다.

#### < 함수 설명 2.>

```
bool Menu::path_check() { // 경로가 올바른지 확인
    ifstream file(path); // 입력받은 경로로 파일을 열어보기
    if(!file.is_open()) { // 파일이 열리지 않을 시 반복
        enter();
        cout << "올바른 경로를 입력해주세요." << endl;
        return true; // 반복문 반복
    }
    return false;
}
```

설명) 경로가 올바른지 확인하는 함수다.

입력받은 경로로 파일을 열고 파일이 열리지 않을 시 반복문을 반복하여 다시 경로를 입력받는다. 잘 열린다면 다음 단계로 넘어간다.

### < 함수 설명 3.>

```
void Menu::STEP_3_print() { // STEP 3. 안내문 출력
    enter();
    next_page();
    enter();
    cout << "미리 준비한 파일의 경로를 입력해주세요." << endl;
    cout << "경로에 한글이 포함될 경우 문제가 발생할 수 있습니다." << endl;
    enter();
}
```

설명) 그냥 안내문을 출력한다.

### < 함수 설명 4.>

```
void Menu::STEP_3() { // STEP 3 실행 함수
    STEP_3_print();
    while(1) {
        cout << "파일 경로를 입력하세요: ";
        input_path(); // 경로를 입력받는다.
        if(path_check()) { // 입력한 경로가 올바른지 않다면
            continue; // 다시 입력받는다.
        }
        break; // 올바르면 그만둔다.
    }
    open_file();
}
```

설명) 파일 경로를 입력해 달라는 안내문과 함께 경로를 입력받고 올바른지 확인까지 하는 함수이다.  
위 함수들을 모두 합해놓은 함수다. main구문을 짧게 줄이기 위해서 만들었다.

### < 함수 설명 5.>

```
void Menu::open_file() { // 한 줄씩 출력하는 함수
    string line; // 한줄 저장할 string 변수
    ifstream file(path); // 파일 경로 열기
    while (getline(file, line))
        cout << line << endl; // 한 줄씩 출력
    file.close(); // 파일 닫기
}
```

설명) 한 줄씩 출력하는 함수다.

파일 저장할 string 변수를 선언한다. 파일 경로를 열고 선언한 변수에 한 줄씩 복사하여 출력 후 파일을 닫는다.

## < 메인 구문 >

```
int main() {
    Menu menu;

    // STEP 1 실행
    menu.print_1();
    while(menu.check_number_ver1()) {} // STEP 1에서 잘못 입력한 경우 반복
    if(menu.number == 1) { // 1 선택 - STEP 2로 진행
        // STEP 2
        menu.STEP_2(); // STEP 2 안내문 출력
        while(menu.check_number_ver3()) {} // STEP 2에서 잘못 입력한 경우 반복
        // STEP 3
        menu.STEP_3(); // STEP 3 실행
    }
    else if (menu.number == 2) { // 2 선택 - 세부기능 2)로 진행
        menu.print_2(); // 세부기능 2)의 안내 메시지 출력
        // 세부 기능 2)실행
        while(true) {
            cout << "'/add' 또는 번호를 입력하세요. ";
            cin >> menu.input;
            if(menu.check_add()) // add 입력시
                continue; // 다시 처음으로
            menu.check_input(); // int형인지 확인하고 int형 변환
            if(menu.check_number_ver2()) { // 알맞은 번호 입력시 STEP 2 실행
                // STEP 2
                menu.STEP_2();
                while(menu.check_number_ver3()) {} // STEP 2에서 잘못 입력한 경우 반복
                // STEP 3
                menu.STEP_3(); // STEP 3 실행
            }
        }
    }
    return 0;
}
```

\* 전반적인 움직임은 아래와 같다.

우선 각 STEP에 따른 안내문과 선택사항을 사용자에게 출력하고 입력을 받는다.  
위 과정에서 올바른 선택을 할 경우 다음 과정으로 넘어가게 된다.  
올바르지 않은 선택을 할 경우 재입력 경고문과 함께 입력 과정을 반복하게 된다.

선택지를 제대로 선택하면 넘어가고 그렇지 않은 경우 순회하는 단순 선택 프로그램을 구현한 것이다.



## 테스트 결과

### STEP 1. 시작 단계

사용자가 접근하기 쉽도록 길잡이가 필요하였고 이에 메뉴의 기능이 필요하다.

세부 기능 1)

선택하는 메뉴창을 띄운다. ( 1. 종합적으로 계산 / 2. 분할적으로 계산 )

성장치를 측정하는 프로그램입니다.  
아래의 선택사항 중 하나를 번호로 입력하세요.

1. 종합적으로 계산
2. 분할적으로 계산

번호를 입력하세요. :

선택 1) 바로 **STEP 2)**로 넘어간다.

번호를 입력하세요. : 1

————— < NEXT PAGE > —————

주기를 선택하세요.  
1달 주기로 분석을 원하시면 1번, 1주 주기로 분석을 원하시면 2번을 선택해주세요.

1. 달
2. 주

번호를 입력하세요. :

선택 2) 밑의 세부 기능 2)로 넘어간다.

2. 분할적으로 계산

번호를 입력하세요. : 2

————— < NEXT PAGE > —————

아래의 선택사항 중 하나를 번호를 입력하세요.  
추가하고 싶은 목록이 있다면 '/add'를 입력하세요.

1. 등
2. 가슴
3. 어깨
4. 하체
5. 이두
6. 삼두
7. 복근

'/add' 또는 번호를 입력하세요. /add

세부 기능 2)

메뉴창을 출력한다. ( 1. 등 / 2. 가슴 / 3. 어깨 / 4. 하체 / 5. 이두 / 6. 삼두 / 7. 복근 )  
사용자가 올바른 번호 입력을 마치면 **STEP 2)**로 넘어간다.

'/add' 또는 번호를 입력하세요. 1

1. 등(을/를) 선택하셨습니다.

주기를 선택하세요.

1달 주기로 분석을 원하시면 1번, 1주 주기로 분석을 원하시면 2번을  
선택해주세요.

1. 달

2. 주

번호를 입력하세요. : █

세부 기능 2-1)

추가 기능을 생성한다. ( 입력 : /추가 )

만일 메뉴의 목록 중 자신이 추가하고자 하는 목록이 있는 경우 추가를 할 수 있도록 도와준다.

ex) test를 추가하고자 한다. ( 입력: /추가 -> 입력 : test )

( 1. 등 / 2. 가슴 / 3. 어깨 / 4. 하체 / 5. 이두 / 6. 삼두 / 7. 복근 / 8. test )

'/add' 또는 번호를 입력하세요. /add

한글이 깨지기 때문에 영어로 입력바랍니다.  
추가할 항목을 입력하세요 : test

아래의 선택사항 중 하나를 번호를 입력하세요.  
추가하고 싶은 목록이 있다면 '/add'를 입력하세요.

1. 등

2. 가슴

3. 어깨

4. 하체

5. 이두

6. 삼두

7. 복근

8. test

'/add' 또는 번호를 입력하세요. █

세부 기능 3)

그 외의 입력일 경우 경고문을 보내고 다시 입력으로 돌아온다.

아래의 선택사항 중 하나를 번호를 입력하세요.  
추가하고 싶은 목록이 있다면 '/add'를 입력하세요.

1. 등
2. 가슴
3. 어깨
4. 하체
5. 이두
6. 삼두
7. 복근
8. test

'/add' 또는 번호를 입력하세요. 9

다시 입력하세요.  
'/add' 또는 번호를 입력하세요. 0

다시 입력하세요.  
'/add' 또는 번호를 입력하세요. a

다시 입력하세요.  
'/add' 또는 번호를 입력하세요. █



## STEP 2. 단위 선택 단계

사용자의 입력을 받기 전에 단위를 선택한다. ( 1. 달 / 2. 주 )

요일마다 받는 것은 수치가 굉장히 많아진다.

또한 근성장은 곧바로 일어나지 않기 때문에 큰 단위를 사용하여 입력받기 위함이다.

세부 기능 1)

단위를 입력받는다.

< NEXT PAGE >

주기를 선택하세요.  
1달 주기로 분석을 원하시면 1번, 1주 주기로 분석을 원하시면 2번을 선택해주세요.

1. 달

2. 주

번호를 입력하세요. :

세부 기능 2)

그 외의 입력일 경우 경고문을 보내고 다시 입력으로 돌아온다.

번호를 입력하세요. : 3

다시 입력하세요.  
번호를 입력하세요. : a

다시 입력하세요.  
번호를 입력하세요. :

### STEP 3. 사용자 입력 저장

사용자가 입력한 수치를 저장한다.

하지만 사용자가 입력법을 모르기 때문에 안내문을 출력한다.

세부 기능 1)

안내문 출력 ( 대충 어떻게 입력하라는지 알려주는 내용 )

```
┌────────────────── < NEXT PAGE > ───────────────────┐  
  
미리 준비한 파일의 경로를 입력해주세요.  
경로에 한글이 포함될 경우 문제가 발생할 수 있습니다.  
  
파일 경로를 입력하세요: █
```

세부 기능 2)

경로를 입력받고 출력하기 ( 경로를 입력받고 해당 파일을 출력한다. )

```
파일 경로를 입력하세요: C:/test_data_week2.csv  
Leg,Shoulder,Back,Chest  
6200,3027,4875,5407  
6540,3777,5507,6790  
PS C:\CPP2409-P> █
```

세부 기능 3)

만일 경로를 읽을 수 없는 경우에는 경고문을 보내고 다시 입력으로 돌아온다.

```
파일 경로를 입력하세요: df  
  
올바른 경로를 입력해주세요.  
파일 경로를 입력하세요: █
```

## 계획 대비 변경 사항

### - 이전 : STEP 3. 사용자 입력 저장 中

세부 기능 2)

사용자로부터 입력 받기

세부 기능 3)

만일 숫자가 아닌 다른 값이 입력된다면 경고문을 출력하고 다시 입력으로 돌아온다.

### - 이후 : STEP 3. 사용자 입력 저장 中

세부 기능 2)

경로를 입력받고 출력하기 ( 경로를 입력받고 해당 파일을 출력한다. )

세부 기능 3)

만일 경로를 읽을 수 없는 경우에는 경고문을 보내고 다시 입력으로 돌아온다.

### - 변경 사항과 사유

직접 입력하여 수치를 저장하는 방식에서 파일을 읽어 수치를 저장하는 방식으로 변경.

우선 사용자의 직접입력은 상당한 시간 소요가 발생하므로 변경하기로 했다. (단점 보완)

## 프로젝트 일정

(표 참고)

| 업무        | 11/3 | 11/10 | 11/13 | 11/16 | 11/23 | 11/30 | 12/2 |
|-----------|------|-------|-------|-------|-------|-------|------|
| 제안서<br>작성 | 完    |       |       |       |       |       |      |
| STEP 1    |      | 完     |       |       |       |       |      |
| STEP 2    |      |       | 完     |       |       |       |      |
| STEP 3    |      |       |       | 完     |       |       |      |
| STEP 4    |      |       |       |       |       |       |      |
| STEP 5    |      |       |       |       |       |       |      |
| 최종 점검     |      |       |       |       |       |       |      |