

**\*\* 반복문, 선택문, 결과 전달 방법으로 문제에서 지시한 것과 다른 방법을 사용할 경우에는 해당 문제에 배당된 점수에서 20%를 감점한다.**

1. (30점) 아래의 코드를 기반으로 행렬을 파일로 저장하고, 행렬의 값을 파일로부터 읽어서 그 합을 화면에 출력하는 프로그램을 작성한다.

```
int main() {
    string filename;
    int row, col;
    get_data(filename, row, col);

    ofstream fout(filename);
    generate_matrix(fout, row, col);
    fout.close();

    ifstream fin(filename);
    print_sum_of_elements(fin);
    fin.close();

    return 0;
}
```

- A. [그림 1.1]에서 보는 것처럼 get\_data() 함수는 키보드로부터 파일명(filename), 행렬의 행(row), 열(column)값을 입력으로 받는다. (8점)
- B. [그림 1.2]에서 보는 것처럼 generate\_matrix()는 아래와 같은 특징을 가지는 row × column 크기의 행렬을 파일에(파일명:filename)에 저장한다. (**for**문을 사용, 저장시 setw(5)를 사용할 것) (12점)

$$a_{i,j} = \begin{cases} [0.100] \text{ 범위의 정수형 랜덤 변수,} & i = j \\ 0, & i \neq j \end{cases}$$

- C. [그림 1.1]에서 보는 것처럼 print\_sum\_of\_elements()는 파일의 모든 원소  $a_{i,j}(i = j)$ 의 총 합을 구해서 그 결과를 화면에 출력한다. (10점)

```
Enter the file name: result.txt
Enter the row and column numbers: 5 3

Sum of all elements: 198
계속하려면 아무 키나 누르십시오 . . .
```

그림 1.1. 출력창

result.txt - 메모장			
파일(F)	편집(E)	서식(O)	보기(V)
41	0	0	
0	85	0	
0	0	72	
0	0	0	
0	0	0	

그림 1.2. 출력 파일

2. (30점) [그림 2]에서 보는 것처럼 줄단위로 text를 입력받아서 하나의 string으로 결합하고 결합된 string에서 주어진 패턴(예: test)이 독립적으로 나타나는 횟수를 계산하여 출력하기 위한 다음 3 함수를 작성한다.

- A. void ReadText(결과 string) : 키보드로 부터 line 단위로 text를 입력받아서 하나의 string으로 결합한다. 종료를 위해서는 빈라인(문자 입력 없이 return)을 입력한다. 결합할 때 현재 줄의 마지막 단어와 다음 줄 첫 단어가 하나로 결합되지 않도록 한다. 결과 string은 pass by reference로 전달한다. (9점)

```
C:\WINDOWS\system32\cmd.exe
Enter line(Hit return for end) 1 : this is test
Enter line(Hit return for end) 2 : etest is not right test
Enter line(Hit return for end) 3 : xtestx is not test
Enter line(Hit return for end) 4 :

%% Combined String Output %%
this is test etest is not right test xtestx is not test

number of matches for [ test ] :3
계속하려면 아무 키나 누르십시오 . . .
```

그림 2. 출력창

- B. int CountCompleteWord(string text, string subStr): text에서 subStr에 있는 패턴이 독자자적으로 존재하는 횟수를 계산하여 return한다. ([그림 2]에서 보는 것처럼 test가 하나의 단어로 나타나는 경우만 고려한다) 반복문은 while문을 사용한다. (12점)
- C. main() : A와 B에서 작성함 함수를 이용하여 [그림 2]와 같은 결과를 출력한다.(9점)

3. (40점) [그림 3]과 같이 정수 수식을 읽어서 결과를 출력하고, 연속적으로 수식을 읽어서 수식계산결과의 합을 출력하기 위한 다음 4 함수를 작성한다.
- bool ReadEquation(피연산자1, 연산자, 피연산자2): 키보드에서 2개의 피연산자와 하나의 연산자(char Type)을 읽어서 pass reference로 결과를 전달한다. 입력된 연산자나 피연산자가 지정된 형이 아니거나 읽기 오류가 발생하면 0을 리턴하고 정상입력이면 1을 리턴한다. 필요할 경우 buffer나 오류 flag를 초기화 한다. (10점)
  - bool Evaluate(피연산자1, 연산자, 피연산자, 결과); 수식을 입력받아서 수식을 계산하고 결과를 pass by reference로 전달한다. 선택문은 Switch문 사용한다. (10점)
  - bool Accumulate(int num, 결과): ReadEquation()을 이용하여 수식을 읽고 Evaluation()을 이용하여 수식을 계산해 결과를 누적하는 작업을 num 횟수만큼 반복한다. 단 수식 read오류가 발생하거나 정상이 아닌 연산자나 피연산자 때문에 수식계산 오류가 발생할 경우에는 수식입력을 중단하고 0을 리턴한다. 단 반복문은 do while문을 사용한다. (10점)
  - Main(): 위에서 작성한 함수들을 사용하여 그림 3.1과 3.2와 같은 결과를 출력한다. 이 함수는 단일 수식을 입력받아 결과를 출력하는 부분과 3개의 수식을 연속으로 읽어서 수식결과의 합을 구해서 출력하는 부분으로 이루어진다. 정상적인 수식들이 입력되면 [그림 3.1]과 같은 결과를 출력하고 비정상적인 수식들이 들어가면 [그림 3.2]와 같은 결과를 출력한다. (10점)

[힌트]

- cin의 읽기 오류 발생시 cin.fail()은 true 를 반환
- cin 버퍼, 오류플래그를 초기화 하기 위해서는 아래의 구문을 사용:  
cin.clear();  
cin.ignore(INT\_MAX, '\n');
- 한줄의 문자열을 키보드로부터 입력 받기 위해서는 아래의 구문을 사용  
char buff[300];  
cin.getline(buff, 300);
- char array(예: char buff[300])형 문자열의 길이를 반환하는 함수: strlen(buff)

```
C:\WINDOWS\system32\cmd.exe
정수계산을 위한 수식을 입력(예: 12 + 13)) --> 30 + 10
30 + 10 = 40

### loop을 이용하여 수식을 3번 입력받아 계산하고 그 결과의 합을 출력 ###
정수계산을 위한 수식을 입력(예: 12 + 13)) -->20 - 10
정수계산을 위한 수식을 입력(예: 12 + 13)) -->10 * 10
정수계산을 위한 수식을 입력(예: 12 + 13)) -->100 / 10

Total = 120
```

그림 3.1 출력창

```
C:\WINDOWS\system32\cmd.exe
정수계산을 위한 수식을 입력(예: 12 + 13)) -->xx + 10

%% Read Error %%

### loop을 이용하여 수식을 3번 입력받아 계산하고 그 결과의 합을 출력 ###
정수계산을 위한 수식을 입력(예: 12 + 13)) -->10 - 5
정수계산을 위한 수식을 입력(예: 12 + 13)) -->xx * 10

%% computation interrupted
계속하려면 아무 키나 누르십시오 . . .
```

그림 3.2 출력창