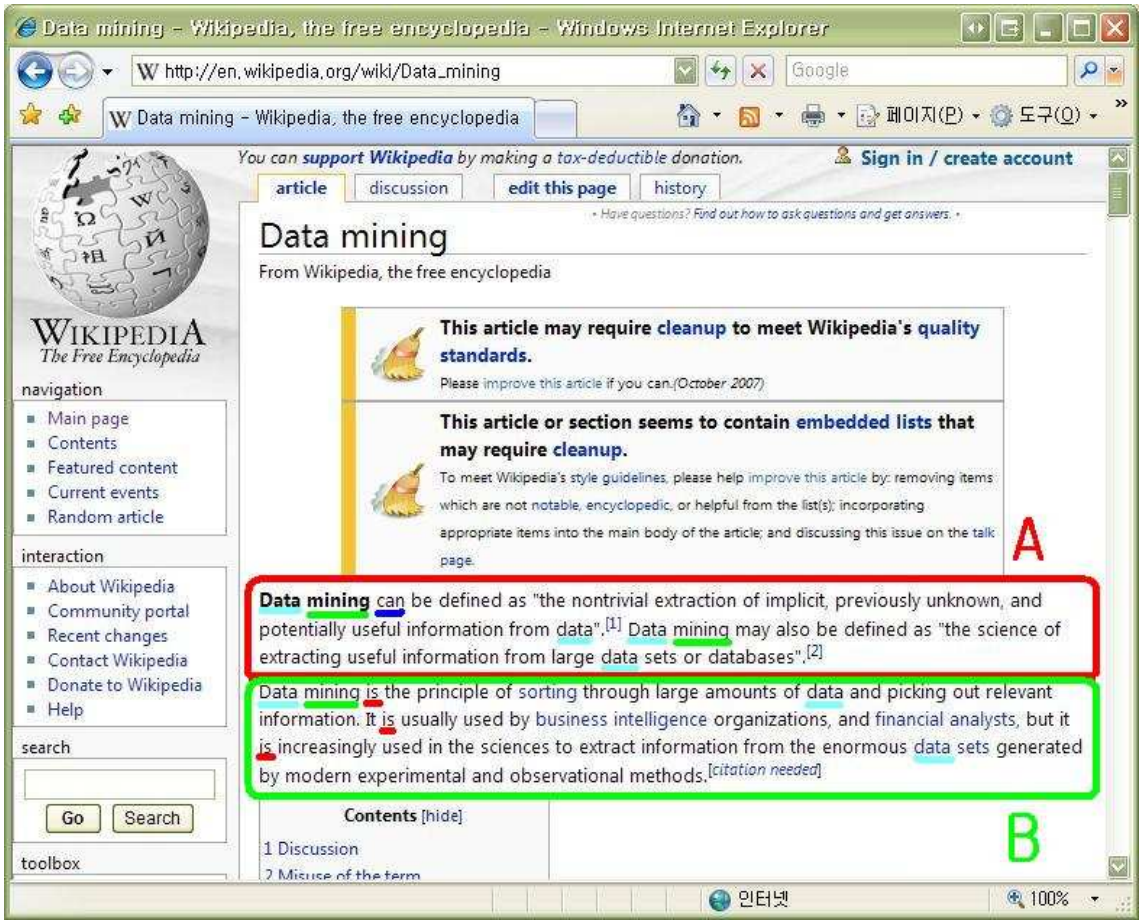
Programming Methodology
Instructor: Kyuseok Shim

Project #2: Counting words in text and +/- operation
Due Date: 0:0 a.m. between 2007-10-22 & 2007-10-23

Introduction

단어 단위로 글을 분해하여 각 단어들이 나온 회수를 셀 수 있다. 예를 들어 아래와 같은 문서에서 A문단에서는 data : 4회, mining : 2회, can : 1회, be : 2회 … 등의 단어가 존재한다. B문단에서는 data : 3회, mining : 1회, is : 3회 … 등의 단어가 존재한다.



문단에 대해서 +,- 연산을 정의하자. A + B는 각 문단에 나오는 단어들의 합집합을 의미한다. 정확하게는 단순히 합집합이 아니라 출현횟수를 더 하는 연산이 된다. 단순한 합집합이라면 A + B를 했을 경우 'data'라는 단어가 있다는 정보만 존재하겠지만 여기서는 data가 7회 문서에서 나타났다는 정보까지 가지고 있는 것이다. 즉, A + B를 하게 되면 data : 7회, mining : 3회, can : 1회, be : 2회, …, is : 3회,… 등이 되는 것이다

이와 유사하게 A – B는 각 문단에서 나오는 단어들의 차집합을 의미한다. 이 또한 출현횟

수를 빼는 연산을 수행하여 한다. 단 결과가 음수가 될 경우에는 그 단어는 출력하지 않는다. 즉, A – B를 수행하면 data : 1회, mining : 1회, can :1회, be :2회… 등이 되고, B – A를 수행하면, 'data'와 'mining', 'can', 'be' 등의 출현 회수가 음수가 되므로 결과에 이 단어들은 표현되지 않고 is : 3회… 등이 결과로 나타나게 된다.

Implementation

1. 문서 파일 내에 있는 단어들의 출현 회수를 구하는 함수를 구현한다.
   대소문자 차이로 인해 다른 단어로 인식되는 것을 방지하기 위해 모든 단어는 대문자로 바꾸어 표시하고, '.'(온점)과 ',(쉼표), '₩n'(줄바꿈), '₩t', ' '(공백)의 특수문자는 delimiter로 취급한다. 나머지 특수문자들은 나오지 않는다고 생각하다. 예를 들어 "I am a student"와 같은 문장을 분석할 경우 I | AM | A | STUDENT 로 분석이 된다. Mr.Kim, I'm 등과 같은 표현은 나타나지 않는다고 가정한다.

2. operator overloading을 통해 정의한 class에 대한 +,- 연산자를 구현한다.
   class를 정의하여 그 class에 '+'와 '-' operator overloading을 하여 두 변수 사이의 +, - 연산이 위에서 언급한대로 이루어지도록 한다.

프로젝트에서는 class 하나와, 그 class내에 text를 읽는 함수, +,- operator는 반드시 구현하여야 한다. 다음의 skeleton code을 참고하도록 한다
※클래스 이름, 함수 이름, 함수 인자 종류 등은 임의로 변경가능

```
// 클래스
class Text {
public :
      // 단어들의 회수가 저장될 구조체
      Map<stirng,int> wordsCount;
      // 생성자
      Text();
      // 문서 파일을 읽어 단어를 세는 함수
      void readText(char* filepath);
      // operator overloading
      Text operator+ (Text t1, Text t2);
      Text operator- (Text t1, Text t2);
      // 결과 출력
      void printAll();
};
```

Input

파일은 다음과 같이 문자들로 이루어져 있다.

text1.txt :

Data mining can be defined as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data". Data mining may also be defined as "the science of extracting useful information from large data sets or databases".

text2.txt :

Data mining is the principle of sorting through large amounts of data and picking out relevant information. It is usually used by business intelligence organizations, and financial analysts, but it is increasingly used in the sciences to extract information from the enormous data sets generated by modern experimental and observational methods.

Output

두 파일 A, B의 단어 분석결과, A+B의 결과, A-B의 결과, B-A의 결과 4가지를 파일로 출력하도록 한다.

>wordscount.exe Text1.txt Text2.txt output.txt

==Text 1.txt==

data : 4

mining : 2

can : 1

be : 2

…

==Text 2.txt==

data : 3

mining : 1

is : 3

…

==Text1.txt + Text2.txt==

…

==Text1.txt – Text2.txt==

…

==Text2.txt – Text2.txt==

…

참고

단어 문단에서 나타난 단어를 저장하기 위해서 STL의 map 클래스를 사용하면 손쉽게 프로그래밍 할 수 있을 것이다.

(사용 방법은 http://www.cprogramming.com/tutorial/stl/stlmap.html 를 참조)

```
// 클래스 생성
map<string,int> mymap;  // 문자열 키 => 정수 값

// 입력
mymap[“Iam”] = 1;
mymap[“a”] = 2;
mymap[“boy”] = 3;

// 사용
cout << mymap[“boy”]; // 3을 출력
```

프로그램을 커맨드라인에서 실행시킬 때 프로그램에 인자를 입력할 수 있다. 구현한 프로그램을 실행하기 위해서는 2개의 문서 파일을 필요한데 이것을 프로그램에 입력해두는 것이 아니라 실행할 때 받을 수 있도록 하는 것이다. 이를 위해서 프로그램의 메인 함수를 다음과 같이 정의한다.

```
...
int main(int argc, char* argv[]) {
...
}
```

argc는 프로그램을 실행할 때 몇 개의 인자들이 커맨드라인에 입력되었나를 나타내는 변수이고, argv[]는 그것들을 문자열로 저장하고 있는 곳이다. 인자들은 빈칸으로 구분된다. 예를 들어 구현한 프로그램의 실행파일이 wordscount.exe일 때,

> wordscount.exe text1.txt text2.txt 100 0.99

를 입력하였다면 argc에는 5 (“명령어”, “text1”, “text2”, “100”, “0.99”)가 argv[0] 에는 “wordsCount.ext”, argv[1]에는 “text1.txt”, …, argv[4]에는 “0.99”가 문자로 저장된다.

이와 같이 main함수에 인자를 줌으로써 파일을 실행할 때, 필요한 값을 입력할 수 있다.

프로그램 실행 예시

다음과 같이 프로그램을 실행할 수 있도록 구현한다.

> 실행파일 문서1 문서2 출력파일

```
text1.txt
   I love you.
text2.txt
   You love me?

> wordscount.exe  text1.txt  text2.txt  output.txt
==Text 1.txt==
I : 1
LOVE : 1
YOU : 1
==Text 2.txt==
YOU : 1
LOVE : 1
ME : 1
==Text1.txt + Text2.txt==
I : 1
LOVE : 2
YOU : 1
ME : 1
==Text1.txt – Text2.txt==
I : 1
==Text2.txt – Text2.txt==
ME : 1
```

제출사항

1. 제출 방법 및 내용

   1) 프로젝트 구현 파일 제출

   - 파일은 ee 홈페이지에 과제 제출을 이용하여 제출한다.
   - 소스 파일과 컴파일을 할 수 있는 파일과 실행파일을 압축하여 제출한다.
   - 각 제출 파일 이름은 다음의 형식을 따른다.

     pro_2_학번.zip

     예) pro_2_2007-12345.zip

- 각 소스 파일 상단에 **E-mail주소와 작성자 학번 이름을** 적는다.

2) 보고서 제출

간단한 설명과 discussion을 하드카피로 제출한다.

2. 제출 일시

파일 제출 마감일 : 2007년 10월 22일과 23일 사이 새벽 0:00

- 구현 파일 제출 시각은 홈페이지 과제 제출 시각에 준한다

보고서 제출 마감일 : 2007년 10월 23일 오전 12:00(정오)

302동 516-2호 앞 과제제출함

채점 규칙

몇 가지 다양한 문서들에 대해서 프로그램이 정확히 동작하는 지 확인하고 operator overloading이 제대로 구현되었는지 확인하여 채점

딜레이는 하루에 10%씩 감점

Copy 발견 시 모든 숙제 0점 처리