

LAB 1: Basic Exercise(기초 연습)

*주의: program에 “이름, 학번, 프로그램ID(ex. Lab1) 표기할것.

<<Lab1-1: File I/O 연습>>

다음 데이터 파일에서 한 line씩 읽고 각 Line의 단어 개수를 출력하는 프로그램을 작성하시오 (특수 기호 무시(, * 등))

예) [Data File] 다음 데이터 파일 이름을 “lab1-1.dat”로 저장.

A program that translates a high level language to a machine language is called a compiler. A compiler is thus a somewhat peculiar sort of program and its output is yet another program.

[출력] 실행했을 경우, 다음과 같은 결과를 출력한다.

A program that translates a high level language to a machine language

The number of words : 12

is called a compiler. A compiler is thus a somewhat peculiar sort of

The number of words : 13

program and its output is yet another program.

The number of words : 8

Total Number of Words: 33

*** 참조: Word Count알고리즘**

```
Void main() {
```

```
...
```

```
while (infile.getline(buffer, 80)) {    // buffer is not empty
```

```
    Print One Line;
```

```
    WordCount (buffer, WC);
```

```
    Print Count (WC);    total = total +wc;
```

```
}
```

```
Print TotalWordCount;
```

```
// end of main
```

Procedure WordCount(buffer, WC) {

//Initialize pointer IN=1, OUT=0; //단어가 시작시, state=IN

int i = 0; int state = OUT; wc=0 //wc=word count

while (buffer[i] != '\0')) { //Until End of Line

if (isalpha(buffer[i])) {

if (state == OUT) {

++wc;

state = IN;

} //문자이고 state=IN이면 do nothing

else if (buffer[i] == ' ')

state = OUT;

i++;

}

● C++ File IO (Lab0 참조)

#include <iostream>

#include <fstream> // for ifstream, open

#include <stdlib.h> // for exit

using namespace std;

void main() {

ifstream infile; // file pointer 정의

char buffer[80]; //buffer size 80

infile.open("lab1.txt", ios::in);

if (infile.fail()) { cout << "can't open the input file" << endl; exit(1); }

.....

while (infile.getline(buffer, 80)) { //buffer not empty 이면 계속

.....

}

Lab1-2: Find Min and Max numbers

1) Data file: 10개의 정수로 구성한다.

예/: 90 40 60 10 50 80 20 70

*2) Output: Minimum number is 10 at position 4,
Maximum number is 90 at position 1*

3) 알고리즘 참조

```
int main(){
    int number, count, data[10];    // 10개 데이터일 경우
    ifstream infile;    int position; // min, max의 위치저장
    while (infile >> number) { //데이터를 파일에서 읽고
        data[count++] = number;    //배열 data에 저장
    }
    .....
    int minnum = findMin(data, count); //find min
    int maxnum = findMax(data, count); // find max
    .....
} //end of main

int findMin (int data[], int n) {
    int min = data[0];           // first data
    for (int i = 1; i < n; i++)
        if (data[i] < min) {
            min = data[i];
            save min position;
        }
    return min;}

```