

COMP319 Algorithms 1, Spring 2021

Homework Programming Assignment 1 (HW1)

알고리즘1, 2021년 봄학기, 프로그래밍 숙제 2

Instructor: Gil-Jin Jang Email: gjang@knu.ac.kr
School of Electronics Engineering, Kyungpook National University
장길진, 경북대학교 전자공학부

프로그래밍 숙제 2번의 목적은 다음과 같다.

1. Linux command line interface (CLI) 에서 argument 기반 compile and execution 실습
2. 파일 입출력 실습
3. 지적재산권 보호를 위한 학번과 이름 표시
4. char array를 이용한 문자열(string)의 입출력/비교/복사 등 기본적인 연산을 실습
5. (핵심) 문자열 정렬 알고리즘 구현(insertion, selection, bubble sort)

1 Homework 2-1

할것들: 주어진 텍스트 파일에서 단어(word)들을 읽어들이어 오름차순으로 정렬하고, 이를 다시 텍스트 파일로 저장하는 프로그램을 작성한다.

1. insertion sort
2. selection sort
3. bubble sort

실행예: 채점에 사용되는 command line argument 는 방법/입력파일/출력파일이다. 실행예는

```
./hw2-1.exe 3 in1.txt out213-3.txt
```

1. 방법(argv[1]): 1 - insertion sort, 2 - selection sort, 3 - bubble sort 를 실행한다.
2. 입력파일(argv[2]): 입력파일은 단어의 개수와 여러 개의 character로 구성된 연속된 단어(word) 들이며, ' ' (공백), '\t' (탭), '\n' (줄바꿈) 으로 구분된다. in1.txt, in2.txt, in3.txt, in4.txt 참조
3. 출력파일(argv[3]): 입력파일과 같은 형식으로 저장되며, 정렬되어 있어야 한다.

입력파일예: 첫번째 줄은 단어의 수이고 단어의 개수에서 제외. 마지막의 숫자 '2020'도 한 단어로 센다.

FILE: in1.txt

41

str replace old new max parameters old this is old substring to be replaced new
this is new substring which would replace old substring max if this optional
argument max is given only the first count occurrences are replaced c 2020

출력파일예: 입력파일과 같은 형식이지만 알파벳 순서로 정렬되어 있음

FILE: out213.txt

41

2020 are argument be c count first given if is is is max max max new new new
occurrences old old old old only optional parameters replace replace replaced
replaced str substring substring substring the this this this to which would

template 코드 및 입출력예제: template-hw2-1.c, in1.txt, in2.txt, in3.txt, in4.txt, out211.txt, out212.txt, out213.txt (in4.txt는 단어의 수가 4096개보다 많아서 출력파일 이 생성되지 않음. Sorting 방법이 달라져도 결과는 모두 동일함.)

1. 파일 입출력이 구현되어 있으며, print_chararr_2dim(), read_chararr_2dim_textfile(), write_chararr_2dim_textfile() 들을 반드시 사용하여야 함. 쉽게 말하면, "FILL" 로 표시가 되어 있는 3개의 정렬 함수만 구현하면 됨.
2. 추가 함수 작성은 허용됨
3. 단어들은 char A[4096][256] 의 이차원 배열로 저장되며, 이에 따라 단어의 최대 길이는 256, 총 4096개까지 처리할 수 있다. (template 파일 참조)
4. 입력예제 in4.txt 는 단어의 수가 6304개로 4096개를 초과하여 read_chararr_2dim_textfile() 에서 읽기 실패가 일어나며, sorting 및 출력파일 생성이 없음.

제출물: hw2-1.c, 오류 없이 컴파일 가능, 위의 실행예에 따라 실행가능해야 하며, 위의 예제의 형식에 맞는 입/출력 파일을 생성해야 한다.

- 조건들:**
1. int main(int argc, char *argv[]) { ... } 로 main 함수가 작성되어야 하며, 컴파일 명령에 따라 "./hw2-1.exe", 또는 "./hw2-1.out"로 실행된다.
 2. 처음의 주석문에는 본인의 정보가 기재되어야 함.
 3. insertion/selection/bubble sort는 작성되어 있는 코드가 굉장히 많으며, 강의자료에도 있다. 일부러 다르게 작성하기 힘들지만, 다른 학생의 코드와 거의 동일하면 COPY 판정을 받을 수 있다. 따라서, 본인이 기존의 코드를 참조했으면 웹주소, 책이름, 강의자료 차수와 쪽수 등을 주석문으로 넣는다. 만약 다른 학생의 코드와 유사한데 참조정보가 없으면 COPY 판정을 받게 된다.
 4. 한글문자는 코드에 넣지 않는다. 컴파일 오류를 발생시킬 수 있으며, 실행점수를 전혀 못 받을 수도 있음. 이름도 영어로 적는다.

채점기준 및 평가방법 Homework 2-1의 총점은 100점이며, 전체 성적에 3% 반영됨

10점 제출 기본점수

10점 학번/이름 주석문에 정확히 입력

60점 컴파일 오류 없고 실행이 정확한지 평가

- 예제로 주어진 입력과 다른 5가지 입력에 대하여 정확한지 평가. 각 예제별로 문자의 순서가 완전히 동일해야 한다(한 글자도 틀리면 그 입력에 대하여 0점)
- 컴파일 오류시 실행불가로 0점이 주어짐
- 답이 맞더라도 알고리즘을 제대로 구현하지 않으면 해당과제를 하지 않은 것으로 판단하여 실행점수 0점

20점 코드에 대한 평가(조건들 만족. 영문만을 사용/소수점/입출력 등)

×0 COPY 적발시 전체 0점. COPIED/BEING COPIED 모두 해당됨.

[실행예] (제대로 실행되었을 경우에는 화면 출력 없음)

```
$ gcc -W -Wall hw2-1.c -o hw2-1.exe
$ ./hw2-1.exe 1 in1.txt out211-1.txt
$ ./hw2-1.exe 2 in1.txt out211-2.txt
$ ./hw2-1.exe 3 in1.txt out211-3.txt
```

```
$ ./hw2-1.exe 1 in2.txt out212-1.txt
$ ./hw2-1.exe 2 in2.txt out212-2.txt
$ ./hw2-1.exe 3 in2.txt out212-3.txt
$ ./hw2-1.exe 1 in4.txt out214-1.txt
cannot read 6304 words (6304 > 4096)
```

2 Homework 2-2

할것들: hw2-1 은 고정된 크기의 2차원 배열을 사용하기 때문에 문자열을 저장하기 위해 메모리의 낭비가 심하다. 따라서 단어의 숫자가 많아지면 실행이 불가능하다. hw2-2에서는 이 문제를 해결하기 위하여 동적 메모리를 할당하는 방법으로 메모리 사용량을 최소화하는 코드를 작성한다.

실행예: 2-1과 유사함. 실행파일 이름만 다름

```
./hw2-2.exe 3 in1.txt out223-3.txt
```

입력파일/출력파일 형식 및 내용: 2-1과 동일함

template 코드 및 입출력예제: template-hw2-2.c, in1.txt, in2.txt, in3.txt, in4.txt, out221.txt, out222.txt, out223.txt, out224.txt (in4.txt에 대해서도 출력파일 생성됨. Sorting 방법이 달라져도 결과는 모두 동일함.)

1. 입출력이 구현되어 있으며, `print_chararr_dynamic()`, `read_chararr_dynamic_textfile()`, `write_chararr_dynamic_textfile()` 들을 반드시 사용하여야 함. 쉽게 말하면, "FILL" 로 표시가 되어 있는 3개의 정렬 함수만 구현하면 됨.
2. 추가 함수 작성은 허용됨
3. 단어들은 `char **A` 의 이차원 포인터에 저장되며, 공간의 낭비를 없애기 위해서 정확하게 문자열의 개수만큼 `malloc()` 함수로 메모리 생성하여 사용함. 또한 단어의 길이 및 개수의 제한이 없다. (template 파일 참조)
4. 입력예제 in4.txt 는 단어의 수가 6304개이지만 출력이 생성된다.
5. `time.h`에 있는 함수들을 이용하여 실행시간을 측정하는 코드가 삽입되어 있다.

제출물: hw2-2.c, 오류 없이 컴파일 가능, 위의 실행예에 따라 실행가능해야 하며, 위의 예제의 형식에 맞는 입/출력 파일을 생성해야 한다.

- 조건들:**
1. `int main(int argc, char *argv[]) { ... }` 로 main 함수가 작성되어야 하며, 컴파일 명령에 따라 `./hw2-2.exe`, 또는 `./hw2-2.out`로 실행된다.
 2. 처음의 주석문에는 본인의 정보가 기재되어야 함.
 3. insertion/selection/bubble sort는 작성되어 있는 코드가 굉장히 많으며, 강의자료에도 있다. 일부러 다르게 작성하기 힘들지만, 다른 학생의 코드와 거의 동일하면 COPY 판정을 받을 수 있다. 따라서, 본인이 기존의 코드를 참조했으면 웹주소, 책이름, 강의자료 차수와 쪽수 등을 주석문으로 넣는다. 만약 다른 학생의 코드와 유사한데 참조정보가 없으면 COPY 판정을 받게 된다.
 4. 한글문자는 코드에 넣지 않는다. 컴파일 오류를 발생시킬 수 있으며, 실행점수를 전혀 못 받을 수도 있음. 이름도 영어로 적는다.

채점기준 및 평가방법 Homework 2-2의 총점은 100점

10점 제출 기본점수

10점 학번/이름 주석문에 정확히 입력

50점 컴파일 오류 없고 실행이 정확한지 평가

- 예제로 주어진 입력과 다른 5가지 입력에 대하여 정확한지 평가. 각 예제별로 문자의 순서가 완전히 동일해야 한다(한 글자도 틀리면 그 입력에 대하여 0점)

- 컴파일 오류시 실행불가로 0점이 주어짐
- 답이 맞더라도 알고리즘을 제대로 구현하지 않으면 해당과제를 하지 않은 것으로 판단하여 실행점수 0점

10점 실행시간 점수

- 10/5/0 3단계
- 10: 다른 학생들에 비해 월등하게 빠름
- 5: 거의 비슷함
- 0: 너무 오래 걸림
- 약간 주관적이지만 큰 문제없이 알고리즘을 구현하였을 경우, 즉 대부분의 학생들이 5점을 받도록 분포를 보고 결정하겠습니다.

20점 코드에 대한 평가(조건들 만족. 영문만을 사용/소수점/입출력 등)

×0 COPY 적발시 전체 0점. COPIED/BEING COPIED 모두 해당됨.

[실행예] (실행시간 출력)

```
$ gcc -W -Wall hw2-2.c -o hw2-2.exe
$ ./hw2-2.exe 1 in1.txt out221-1.txt
>> Execution time: 0.00002 seconds
$ ./hw2-2.exe 2 in1.txt out221-2.txt
>> Execution time: 0.00001 seconds
$ ./hw2-2.exe 3 in1.txt out221-3.txt
>> Execution time: 0.00002 seconds
$ ./hw2-2.exe 1 in2.txt out222-1.txt
>> Execution time: 0.00007 seconds
$ ./hw2-2.exe 2 in2.txt out222-2.txt
>> Execution time: 0.00005 seconds
$ ./hw2-2.exe 3 in2.txt out222-3.txt
>> Execution time: 0.00016 seconds
$ ./hw2-2.exe 1 in4.txt out224-1.txt
>> Execution time: 0.14915 seconds
```

3 제출형식 및 방법

제출할 코드 hw2-1.c, hw2-2.c 만을 제출해야 함. 필요없는 파일 제출시 10점까지 감점 있음

제출방법 위의 2개의 파일을 묶어서 hw2.zip 을 만들고, lms.knu.ac.kr 에 업로드한다. LMS는 제출된 파일의 이름을 복잡하게 바꾸기 때문에 개별파일을 제출하면 채점이 매우 어렵다.