

COMP319-004 Algorithms, Spring 2020

Programming Assignment 2

Instructor: Gil-Jin Jang (gjang@knu.ac.kr)

School of Electronics Engineering

1. 정수입력에 대한 정렬 알고리즘 구현하기

- 다음과 같이 주어진 입출력 조건에 따라 *삽입(insertion)*, *선택(selection)*, *거품(bubble)* 정렬 알고리즘을 구현한다.
- **구현방법:** `template_sort_int_simple.c` 를 수정해서 작성함

INPUT:

- (텍스트 파일) 숫자의 개수 숫자 숫자 ...

입력예제파일1: `input10`

10

74 21 42 94 13 24 80 45 75 37

입력예제파일2: `input1234` (중간 ... 로 생략)

1234

415 540 556 268 461 142 356 99 573 254 ... 158 63 64 280 584 464 177 128 256 361 564 323

실행예:

- command line interface 일 경우, `code::blocks` 등 GUI는 단축키나 클릭으로 실행 가능하며 키보드 입력 부분은 동일함.
- '?' 이후 부분은 키보드 입력

```
$ sort_int_simple.exe
```

```
Input file name? input10
```

```
Output file name? insertion10
```

```
Method (1: insertion, 2: selection, 3: bubble)? 1
```

```
$ sort_int_simple.exe
```

```
Input file name? input1234
```

```
Output file name? bubble1234
```

```
Method (1: insertion, 2: selection, 3: bubble)? 3
```

OUTPUT: 3가지 방법으로 정렬된 파일

- insertion10, selection10, bubble10, insertion1234, selection1234, ...

출력예제파일1: insertion10

10

13 21 24 37 42 45 74 75 80 94

출력예제파일2: bubble1234 (중간 ... 로 생략)

1234

1 2 2 2 2 3 3 4 4 4 ... 594 595 595 595 595 596 596 596 596 597 598 599

2. 문자입력에 대한 정렬 알고리즘 구현하기

- 다음과 같이 주어진 텍스트파일에 적용할 수 있는 정렬 알고리즘을 구현한다.
- 구현방법: template_sort_chararray_simple.c 를 수정해서 작성함

INPUT:

- (텍스트 파일) 단어의개수 단어 단어 ...
- 단어는 모두 소문자로만 구성됨

입력예제파일1: imagine (140단어로 구성된 노래 가사, 중간 ... 로 생략)

140

imagine there s no heaven

it s easy if you try

no hell below us

...

but i m not the only one

i hope someday you ll join us

and the world will live as one

입력예제파일2: ihaveadream (1682단어로 구성된 연설문, 중간 ... 로 생략)

1682

i have a dream

i am happy to join with you today in what will go down in history

as the greatest demonstration for freedom in the history of our nation

five score years ago a great american

in whose symbolic shadow we stand today

signed the emancipation proclamation
this momentous decree came as a great beacon light of hope to millions of
negro slaves who had been seared in the flames of withering injustice
it came as a joyous daybreak to end the long night of their captivity
...
let freedom ring from stone mountain of georgia
let freedom ring from lookout mountain of tennessee
let freedom ring from every hill and molehill of mississippi
from every mountainside let freedom ring
and when this happens and when we allow freedom ring
when we let it ring from every village and every hamlet
from every state and every city
we will be able to speed up that day when all of god s children
black men and white men
jews and gentiles protestants and catholics
will be able to join hands and sing in the words of the old negro spiritual
free at last free at last
thank god almighty we are free at last

실행예:

- command line interface 일 경우, code::blocks 등 GUI는 단축키나 클릭으로 실행 가능하며 키보드 입력 부분은 동일함.
- '?' 이후 부분은 키보드 입력

```
$ sort_chararray_simple.exe
Input file name? imagine
Output file name? bubbleimagine
Method (1: insertion, 2: selection, 3: bubble)? 3
- bubble sort로 정렬된 bubbleimagine 파일 만들어짐
```

```
$ sort_chararray_simple.exe
Input file name? imagine
Output file name? selectionimagine
Method (1: insertion, 2: selection, 3: bubble)? 2
- selection sort로 정렬된 selectionimagine 파일 만들어짐
```

```
$ sort_chararray_simple.exe
Input file name? ihaveadream
Output file name? insertionihaveadream
Method (1: insertion, 2: selection, 3: bubble)? 1
```

- insertion sort로 정렬된 insertionimage 파일 만들어짐

OUTPUT: 3가지 방법으로 정렬된 파일

- `insertionimage`, `selectionimage`, `bubbleimage`,
- `insertionhaveadream`, `selectionhaveadream`, `bubbleihaveadream`

출력예제파일1: selectionimage (실제 출력에서는 줄바꿈 없음)

140

a a a above ah all all all all and and and as as
be below brotherhood but but can countries die do
dreamer dreamer easy for for for greed hard heaven
hell hope hope hunger i i i i i i i if if
imagine imagine imagine imagine imagine imagine
in isn it it join join kill life live livin livin
ll ll m m m m man may may need no no no no no no not not
nothing of one one one one only only only or or
peace people people people possessions religion
s s s say say sharing sky someday someday t
the the the the the the the the there there to to
today too try us us us us will will wonder
world world world you you you you you you you you

출력예제파일2: insertionihaveadream (중간 ... 로 생략, 실제 출력에서는 줄바꿈 없음)

1682

a
able able able able able able able able again again ago ahead alabama alabama alabama
all all all all all all all alleghenies allow allow almighty alone also always
am am america america america america america american american american american
an an an an and and and and and and and and and and and and and and and and
...
we well
were what when when when when when when where where where
which which which which which whirlwinds white white white white white whites
who who who who whose will will will will will will will will will will will
will will will will will will will will will will will will will will will
winds with with with with with with with with with with with with with with with
withering words words words work work would would wrongful wrote
years years years years years yes york york you you you you you you you your

3. 문자입력에 대한 정렬 알고리즘 구현하기, 방법2

- INPUT, OUTPUT: 2번과 동일
- 구현방법: `template_sort_string_simple.c` 를 수정, `struct String` 을 사용해야 한다.
- 실행예: (command line interface 일 경우, `code::blocks` 등 GUI는 단축키나 클릭으로 실행 가능하며 키보드 입력 부분은 동일함. '?' 이후 부분은 키보드 입력

```
$ sort_string_simple.exe
Input file name? imagine
Output file name? stringbubbleimagine
Method (1: insertion, 2: selection, 3: bubble)? 3
- bubble sort로 정렬된 stringbubbleimagine 파일 만들어짐
```

```
$ sort_string_simple.exe
Input file name? imagine
Output file name? stringselectionimagine
Method (1: insertion, 2: selection, 3: bubble)? 2
- selection sort로 정렬된 stringselectionimagine 파일 만들어짐
```

```
$ sort_string_simple.exe
Input file name? ihaveadream
Output file name? stringinsertionihaveadream
Method (1: insertion, 2: selection, 3: bubble)? 1
- insertion sort로 정렬된 stringinsertionimagine 파일 만들어짐
```

4. 주어진 것들

- 예제 template 소스 파일:
 - `template_sort_chararray_simple.c` `template_sort_int_simple.c`
`template_sort_string_simple.c`
 - 파일 입출력은 모두 구현되어 있으며, `/* FILL */` 로 되어 있는 부분만 채워 넣으면 된다.
 - bubble sort 는 일부 구현되어 있으며 다른 정렬 코드는 이를 참고하여 구현한다
 - 예제 입력 파일:
 - `input10` `input1234` `input58` `imagine` `ihaveadream`
 - 출력 예제 파일(참고용):
 - `insertion10` `bubble1234` `bubbleimagine` `insertionihaveadream` ...
-

5. 제출물

- 코드: 주어진 template 코드들에서 비워져 있는 부분(**/* FILL */**)을 채워 넣는다.
 - `sort_chararray_simple.c` `sort_int_simple.c` `sort_string_simple.c`
 - 생성한 입출력 예제는 제출할 필요없다 소스만 가지고 평가
- 코드 참조방법:
 - insertion, selection, bubble sort 는 매우 잘 알려진 방법들이라 공개된 코드들이 많다.
 - 인터넷에서 검색하여 공개된 코드들을 그대로 사용하여도 되지만 반드시 참조했음을 남겨야 한다.
 - 알고리즘을 보고 직접 작성하였으면 남길 필요는 없다.
 - 만약 다른 사람과 코드가 동일한데 참조했음을 알리는 표시가 없으면 copy로 간주되어 0점 처리될 수 있음
 - (예를 들어 학생 A와 B의 코드가 동일한데 A는 참조를 남겼고 B는 안 남겼으면 A는 감점 없고 B는 0점처리됨)

예: 파일 `template_sort_int_simple.c`

```
void bubble_sort(int arr[], int n) {  
    /* sample reference  
    * source: https://ko.wikipedia.org/ */  
    int i, j;  
    ...  
}
```

6. 배점

1. 학번과 이름(20%): 각 소스코드의 맨 윗줄은 자신의 학번과 이름(중요!!)
2. 구현 적합성(30%): 문제에 주어진 조건 만족. 예를 들어 insertion, selection, bubble sort 알고리즘들이 제대로 구현되어 있어야 한다.
3. 정확성(30%): 주어지지 않은 다른 예제들에 대해서도 정확한 출력 생성
4. 코드(20%): 코드에 대한 평가. 다른 사람이 쉽게 이해할 수 있도록 작성되었는지 평가
5. 제출물(-10%): 3개의 소스코드파일(`sort_chararray_simple.c` `sort_int_simple.c` `sort_string_simple.c`) 이외에 다른 파일이 포함되면 10% 감점함
6. 지각제출(-10%/시간): 1시간당 10% 감점함
7. 부정행위(-100%): 다른 사람의 코드와 동일하면 0점 처리될 수 있다. 외부 참조를 했을 경우 반드시 출처를 남겨야 한다.