

객체지향프로그래밍

Assignment Report 1-3

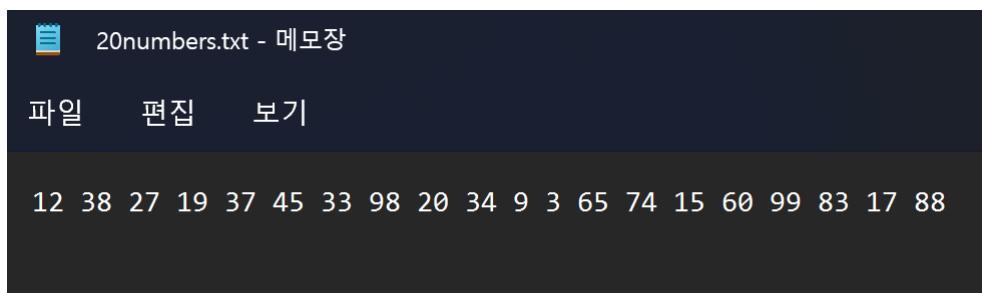
2019202050 이강현

문제 1

파일 "Assignment.txt"로부터 임의의 숫자 20개를 입력 받아 ascending order로 정렬해 출력하는 프로그램을 구현하시오. 이 때 정렬 알고리즘은 자유롭게 채택하며 STL이나 그 외 Built in function 등의 함수를 사용하지 않고 직접 구현한 정렬 알고리즘을 사용하여 구현한다.

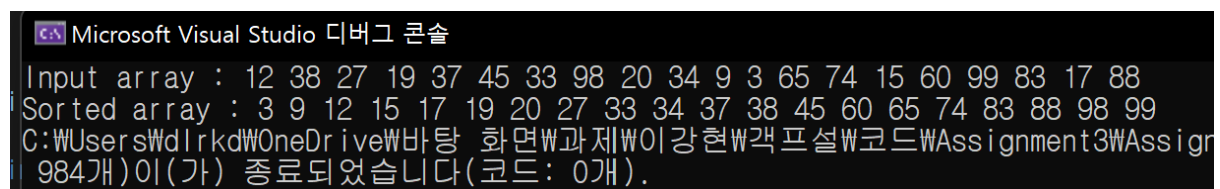
필요한 개념: 메모장에서 입력값을 받아와야 하므로 스트림을 이용할 줄 알아야 한다.

오름차순으로 정렬하는 알고리즘을 알아야 한다.



```
20numbers.txt - 메모장
파일 편집 보기
12 38 27 19 37 45 33 98 20 34 9 3 65 74 15 60 99 83 17 88
```

<text 파일 내 숫자 저장 포맷 예시>



```
Microsoft Visual Studio 디버그 콘솔
Input array : 12 38 27 19 37 45 33 98 20 34 9 3 65 74 15 60 99 83 17 88
Sorted array : 3 9 12 15 17 19 20 27 33 34 37 38 45 60 65 74 83 88 98 99
C:\Users\Wd\OneDrive\바탕 화면\과제\이강현\객체프로그래밍\코드\Assignment3\Assignment3.cpp(984개)이(가) 종료되었습니다(코드: 0개).
```

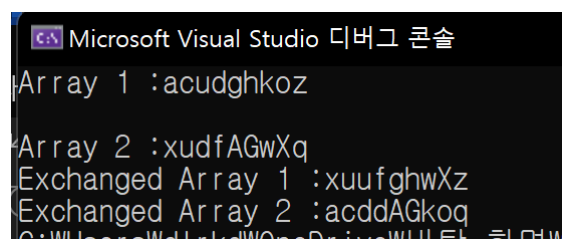
<프로그램 출력 예시>

고찰: 메모장 속 숫자입력을 가져오기 위한 입력 스트림을 만들어주고 배열을 선언하여 20개의 숫자를 배열에 저장해두고 반복문의 중첩을 이용하여 오름차순 정렬 알고리즘을 구현했다. 정렬 방법은 앞에서부터 두개씩 짝을 지어 비교연산을 반복하는 방법이다. 반복문 두개는 변수 i와 p를 사용했는데 i와 p의 관계식을 이용하지 않고 문제를 해결하려다가 모든 경우의 비교연산을 하지 못해 오류가 났었다. 이는 $p=i+1$ 을 이용하여 해결하였다.

문제 2

같은 길이의 2개의 영어 문자열을 입력 받아 비교해서 문자를 이동하는 프로그램을 구현하시오. 프로그램은 두 개의 문자열을 받아서 각 문자열의 문자를 순서대로 비교해 알파벳 순으로 더 뒤에 있는 문자를 첫번째 문자열로, 더 앞에 있는 문자를 두번째 문자열로 이동시키고 문자열의 모든 문자가 비교, 이동이 끝난 후 두 문자열을 다시 출력한다. 이 때 대문자, 소문자는 구별하지 않는다.

필요한 개념: 문자열을 입력받아오기 위한 스트림 형성, 한 문자씩 비교하기 위한 char형 배열을 두개 선언하여 각 자리에 접근하고 비교 연산, 대소문자를 구별하지 않기 위한 조건



```
Microsoft Visual Studio 디버그 콘솔
Array 1 :acudghkoz
Array 2 :xudfAGwXq
Exchanged Array 1 :xuufghwXz
Exchanged Array 2 :acddAGkoq
```

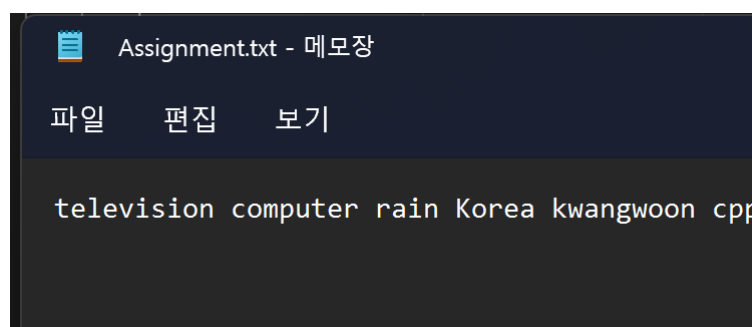
<프로그램 입력 및 출력 예시>

고찰: 배열 두개에 문자단위로 하나씩 입력받아와 값 저장을 한 후 아스키코드를 이용한 비교연산을 진행하였다. 대소문자를 구별하지 않는다는 조건으로 문자간 비교를 할 경우 아스키코드상 소문자 구간을 조건문으로 달아 구간이 다를경우 대문자인 곳에 32를 더해주어 대문자를 소문자로 바꿔주어 연산하였다.

문제 3

파일 "Assignment.txt"로부터 임의의 영단어(최대 100개)를 입력 받아 길이별로 출력하는 프로그램을 구현하시오. 영단어는 최대 20의 길이를 가지며 문자열의 길이는 문자열의 길이를 구하는 함수를 직접 구현해서 구한다.

필요한 개념: 입력 스트림을 형성할 줄 알아야 함. 길이별로 출력해야 하므로 문자열의 길이를 구할 수 있는 방법이 있어야 함.



```
Assignment.txt - 메모장
파일 편집 보기
television computer rain Korea kwangwoon cpp
```

<Text파일 내 영단어 저장 포맷 예시>



```
length3: cpp
length4: rain
length5: Korea
length8: computer
length9: kwangwoon
length10: television
```

<프로그램 출력 예시>

고찰: 문자단위로 2차원배열에 값들을 저장해두고 length[][]라는 2차원배열에 길이 별로 저장한 후 출력하는 과정을 거쳤다. 문자단위로 스트림을 통해 반복문을 이용하여 문자단위로 받아오다가 공백을 만나면 배열을 바꿔 새로운 단어를 받을 수 있게 하였고 공백을 만난 후 null문자를 또 받는다면 반복문을 탈출하게 하는 구조이다. 이러한 구조 때문에 메모장에서 마지막단어를 입력해주고 공백을 넣지 않은채 저장을 하게되면 마지막단어의 길이수가 제대로 나오지 않는 것을 확인하였다. 그래서 문자열을 통째로 받아와 저장을 한 후 공백의 개수+1이 단어 수 이므로 그에 맞게 개별저장을 했다면 메모장에 마지막 단어에 공백을 넣어주지 않아도 정상적으로 출력이 되었을 것 같았다. 하지만 이 방법 또한 메모장에 공백을 많이 넣어준다면 문제가 되는 알고리즘이다. 따라서 공백을 조건으로 넣어주고 문자열의 맨끝에 null문자가 삽입된다는 것을 이용하면 이 문제를 해결할 수 있었을 것 같다.

4. 5x5의 보드에서 Command를 입력 받아 Node가 움직이는 프로그램을 구현하시오. 보드는 Node와 Node의 경로를 제외하고 모두 '0'으로 출력하며 Node는 H, 노드가 움직이기 4번째 전까지의 경로는 x로 출력한다. 노드는 프로그램 시작 시 좌상단의 끝(0,0)에 있으며 아래의 Command에 의해 움직인다.(이 때 보드의 범위를 넘어가는 명령은 무시하며 움직이지 않은 것으로 취급한다.) 프로그램은 시작 시 노드와 보드를 출력하고 매 명령을 입력받을 때 마다 재 출력한다.

필요한 개념: 보드의 형성방법, 보드를 계속 생성하고 조건을 만족하는 입력을 계속 받기위한 반복문 사용, 노드의 자취를 보여줄 수 있는 알고리즘, 범위를 넘어가는 즉, 조건을 만족하지 않는 입력을 해결하기위한 조건등

```
Microsoft Visual Studio 디버그 콘솔
H 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : d
X H 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : d
X X H 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : d
X X X H 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : d
X X X X H
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : s
0 X X X X
0 0 0 0 H
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : a
0 0 X X X
0 0 0 H X
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : a
0 0 0 X X
0 0 H X X
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : w
0 0 H 0 X
0 0 X X X
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
Enter Move Command(a: Left, d: Right, w: Up, s: Down, q: Finish) : q
C:\Users\WdIrk\OneDrive\바탕 화면\과제\이강현\객프설\코드\Assignment3\WA
이 창을 닫으려면 아무 키나 누르세요...
```

<프로그램 실행 결과>

고찰: 25개의 배열을 선언하고 (0,0)의 위치가 첫 노드의 위치이므로 첫번째 배열에는 1 나머지 배열에는 0을 넣어두는 방식으로 초기화하였다. 반복문과 나머지를 이용하여 노드를 5x5형식으로 출력해주고 함수를 이용하여 보드를 벗어나는 입력에 대한 대처와 노드의 움직임을 구현하였다.

자취를 표현하는 알고리즘은 길이 4인 배열과 증가연산을 이용하여 구현하였는데 한가지 문제점은 노드가 보드가 좌우로 왔다갔다 하는 연산과 한바퀴를 가장 짧은 거리로 돌 때 문제가 생겼는데 배열에 현재 노드의 위치를 저장하고 증가시켜 다음 배열에 노드 위치를 저장하며 결국 제일 오래된 배열의 데이터 즉 4번째 자취를 표현한 후 다음연산때 그 위치에 0을 넣어주고 새로운 위치를 덮어쓰우는 형식으로 프로그램을 구현하다보니 좌우로 왔다갔다하거나 한바퀴를 돌 때 자취가 겹치게 되면 1번째 자취와 없어질 타이밍인 자취가 서로 위치가 겹쳐 자취가 끊기는 현상을 목격하였다. 대각선으로 움직이지 않는 이상 노드가 4개의 자취를 표현하며 보드를 움직일 때 자취가 사라지는 타이밍은 첫번째 배열과 마지막 배열이 같을 때를 제외하곤 없었다.

따라서 이는 노드의 위치를 저장하는 배열의 첫번째 배열과 마지막 배열의 값이 같게 되면 해당 자리에 0을 넣어주지 않도록 하는 조건문을 통해 해결하였다.