

2022-1 윈도우 프로그래밍 워밍업 문제-1

2022년도 1학기

1. [1차원 배열 사용하기] 문자열 다루기

- 영어 대소문자와 띄어쓰기만으로 이루어지고 마지막에 마침표로 마치는 문장을 입력받아 1차원 배열에 저장한다.
 - 문장의 길이: 최대 60자 (최대 길이는 변경 가능).
 - 단어는 띄어쓰기로 구분하는데 1개 이상의 연속 공백이 있는 경우는 1개의 공백으로 처리한다.
- 다음의 명령어를 실행한다.
 - e/E: 단어에 e 또는 E자가 있는 경우에는 그 단어를 모두 대문자로 바꾸고 출력
 - f/F: 모든 공백에 랜덤한 문자 (@, %, # 또는 .)를 넣고 출력
 - l/L: 단어의 개수를 출력
 - c/C: 대문자는 소문자로, 소문자는 대문자로 바꾸기
 - a/A: 문장의 문자를 좌측/우측으로 한 문자씩 이동한다.
 - q/Q: 프로그램을 종료

결과 예 1) Input sentence: This is C sample example for window programming.

명령어 e: This is C SAMPLE EXAMPLE for window programming.

명령어 l: Result: 8 words

명령어 c: tHIS IS c sample example FOR WINDOW PROGRAMMING.

명령어 f: tHIS.IS.c.sample.example.FOR.WINDOW.PROGRAMMING.

명령어 f: tHIS IS c sample example FOR WINDOW PROGRAMMING.

명령어 c: This is C SAMPLE EXAMPLE for window programming.

명령어 a: his is C SAMPLE EXAMPLE for window programmingT.

명령어 a: is is C SAMPLE EXAMPLE for window programmingTh.

//--- e 가 있는 단어를 대문자로 바꿈

//--- 단어의 개수 출력

//--- 대문자를 소문자로, 소문자를 대문자로 바꿈

//--- 공백에 . 을 입력함

//--- .을 다시 공백으로 바꿈

//---대문자를 소문자로, 소문자를 대문자로 바꿈

//--- 문자열이 좌측으로 한 칸 이동, 맨 앞의 문자 맨뒤로

//--- 문자열이 좌측으로 한 칸 이동, 맨 앞의 문자 맨뒤로

2. [2차원 배열 사용하기] 4x5 행렬 만들고 행렬의 연산 구현하기

- 4행 5열의 행렬을 만들고 숫자(중복 불가)를 행렬에 저장한다.
 - 1 ~ 50 사이의 랜덤한 숫자를 행렬에 입력
 - 열의 위치를 맞춰서 출력한다.
- 다음의 명령어를 수행한 후 결과를 출력한다.
 - a: 1행1열을 시작으로 행을 따라 오름차순으로 정렬
 - d: 1행5열을 시작으로 열을 따라 내림차순으로 정렬
 - e: 짝수만 출력하기
 - o: 홀수만 출력하기
 - m: 최대값 출력
 - n: 최소값 출력
 - p: 한 행의 값을 모두 더해 각 행의 1열에 저장
 - r: 랜덤하게 재정렬
 - s: 다시 입력받기 시작
 - q: 프로그램 종료

결과 예)

행렬값:

3	10	24	11	18
39	41	5	17	22
1	29	28	13	20
25	14	33	43	50

명령어 입력 a:

1	3	5	10	11
13	14	17	18	20
22	24	25	28	29
33	39	41	43	50

명령어 입력 d:

10	17	24	33	50
5	14	22	29	43
3	13	20	28	41
1	11	18	25	39

명령어 입력 e:

0	10	24	0	18
0	0	0	0	22
0	0	28	0	20
0	14	0	0	50

명령어 입력 m:

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	50

명령어 입력 r:

10	41	18	3	29
5	28	43	25	14
33	17	24	11	50
39	22	13	20	1

명령어 입력 p:

101	41	18	3	29
152	28	43	25	14
135	17	24	11	50
95	22	13	20	1

3. [구조체 사용하기] 텍스트를 사용하여 2인 돌 이동하기

- 화면에 10x10 바둑판을 그린다. 바둑판의 임의의 위치에 2개의 다른 돌을 놓고 사용자의 명령어(wasd 와 ijkl)에 따라 돌을 좌/우/상/하로 번갈아 입력 받아 움직인다.
 - 두 종류의 다른 돌은 번갈아 가며 움직일 수 있다.
 - 두 사용자의 이동 명령어는 두 종류의 다른 조합을 사용한다. (wasd와 ijkl)
- 가장자리에 돌이 닿으면 반대편 가장자리에 나타난다. 반대편에 나타날 때 돌의 모양이 바뀐다.
 - 돌의 모양은 5개 중 한 개로 랜덤하게 선택한다: 5개의 돌 모양 예) o, x, #, @, *
- 움직이는 돌이 다른 돌과 부딪치면
 - 움직인 돌 (위의 돌)의 모양만 나타난다. 이때 비프 소리를 출력한다.
 - 다음 순서인 아래의 돌을 움직이게 되면 바뀐 모양으로 나타난다.
- 돌의 위치와 상태를 가지고 있는 구조체를 사용한다.
- 키보드 명령으로 이동 키보드 외에 프로그램 리셋/종료 명령을 추가한다.
 - s: 프로그램 리셋 (시작위치에서 새로운 돌 모양으로 시작)
 - q: 프로그램 종료

** 비프음 소리내기: 함수 Beep 사용

```
#include <windows.h>
```

```
Beep (음의 높이, 음의 지속시간);
```

결과 예)

```
-----
| | |x| | | | | | |
-----
| | | | |O| | | | |
-----
...
```

Input command: j (x가 좌측 이동)

```
-----
| |x| | |O| | | | |
-----
| | | | | | | | | |
-----
...
```

Input command: a (O가 왼쪽으로 가니 x와 부딪치게 됨. O이 x위에 올려져서 O만 출력된다.)

```
-----
| | | |x|O| | | | |
-----
| | | | | | | | | |
-----
```

Input command: k (다음 순서로 x를 아래로 움직이면, x의 모양이 다른 모양으로 바뀌어 나타난다)

```
-----
| | | |O| | | | |
-----
| | | |@| | | | |
-----
```

Input command: w (원이 위로 이동)

```
-----
| | |x| |O| | | | |
-----
| | | | | | | | | |
-----
...
```

Input command: d (원이 우측으로 이동)

```
-----
| |x| | |O| | | | |
-----
| | | | | | | | | |
-----
...
```

...

.....

- 문제 당 채점 기준
 - 1문제당 3점으로 계산하여 적용됨
 - 각 문제당 90% 이상 구현했을 때: 3점
 - 각 문제당 50% 이상 구현했을 때: 2점
 - 각 문제당 30% 이상 구현했을 때: 1점