

# CSED101. Programming & Problem solving

## Fall, 2020

### Programming Assignment #2

(75 points)

김태훈(taehoon1018@postech.ac.kr)

■ 제출 마감일: 2020.11.18 23:59

■ 개발 환경: Windows Visual Studio 2019

#### ■ 제출물

- C Code files (assn2\_1.c, assn2\_2.c)
  - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- 보고서 파일 (assn2.docx, assn2.hwp 또는 assn2.pdf)
  - AssnReadMe.pdf 를 참조하여 작성할 것.
  - 명예서약(Honor code): 표지에 다음의 내용을 포함한다. “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
  - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

#### ■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 채점을 위해 실행 예시와 최대한 비슷하게 작성해 주세요.
- 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
- 컴파일 & 실행이 안되면 무조건 0점 처리된다.
- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 ‘POSTECH 전자컴퓨터공학부 부정행위 정의’를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

## [들어가기 전]

### 1. 문자열(string)

- 연속된 문자들로 C 언어에서 문자열 앞 뒤에 " "를 이용한다.
- char 형의 1차원 배열을 이용하여 문자열을 저장한다.
- 배열에 문자열을 저장할 때는 끝을 NULL 문자 ('\0')를 넣어서 표시한다.

### 2. 선언

- `char str[] = "hello";`  
위와 같이 선언과 동시에 초기화를 하게 되면, 자동으로 문자열의 끝에 널문자가 추가된다.

str

h	e	l	l	o	\0	
---	---	---	---	---	----	--

### 3. 입력과 출력

```
char str[100];  
printf("Input your POSTECH ID: ");  
scanf("%s", str);  
printf("%s", str);
```

<결과화면> (아래의 파란 글자는 사용자 입력입니다.)

```
Input your POSTECH ID: taehoon1018  
taehoon1018
```

## ■ Problem 1: 행맨 게임 (Hangman Game) (35점)

### [문제]

행맨 (Hangman) 게임은 상대방이 생각하는 단어를 맞히는 게임이다. 컴퓨터가 선택한 단어를 맞히는 행맨 게임을 작성해 보자.

### [목적]

- 1차원 배열의 선언과 사용을 익힌다.
- 함수에서 1차원 배열을 매개변수로 사용하는 방법을 익힌다.
- 텍스트 파일을 통한 입력을 익힌다.

### [주의사항]

- (1) 파일 이름은 "assn2\_1.c"로 저장 한다.
- (2) 보고서는 "assn2.docx", "assn2.hwp" 또는 "assn2.pdf"로 저장 한다. (보고서는 통합하여 작성)
- (3) 전역 변수, goto 문, string 관련 함수, 구조체 등은 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능하다.
- (4) string.h에 정의된 함수를 사용하지 않는다.
  - 해당 기능이 필요한 경우 char 배열을 이용, 직접 구현하여 사용한다.
- (5) 모든 기능을 main 함수에서 구현한 경우 감점 처리한다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 함수를 정의해서 구현하여야 한다.
- (6) 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않는다.
- (7) 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

### [가정]

- (1) 단어의 최대길이는 30 이라고 가정한다.  
#define 지시자를 이용하여 단어의 최대길이를 정의하여 사용한다.
- (2) 단어는 모두 영문 소문자로만 구성되어 있다고 가정한다.
- (3) 모든 입력 값은 영문 소문자만 입력 받는다고 가정한다.

### [설명]

컴퓨터가 dic.txt 파일에 있는 단어 중 하나를 랜덤하게 선택한다. 그리고 아래의 예시와 같이 선택된 단어의 길이만큼 '\_'가 출력된다. 행맨의 생명(life)은 10부터 시작한다. 단어를 맞추기 위해 사용자가 영문자를 하나 입력하는데, 선택된 단어에 입력한 영문자가 없는 경우 life의 수가 1감소한다. 10번의 없는 문자를 입력하여 life 가 0이 되기 전에 단어를 맞춰야 하며 life 가 0이 되면, 선택된 단어를 공개(출력)하고 게임은 종료된다.

선택된 단어에 입력 받은 문자가 있는 경우, 아래의 예시와 같이 해당 부분의 문자만 출력한다. 사용자가 입력한 모든 문자는 Used란에 입력한 순서대로 출력한다. 단, 입력한 문자가 Used란에 있는 경우, life를 감소하지 않고 이미 입력된 값이라고 출력한다.

아래의 예제는 게임이 실행되어, language라는 단어가 랜덤하게 선택된 경우이다. 예시의 빨간색 밑줄은 사용자 입력에 해당한다.

```

Let's play the hangman game!
Word: _ _ _ _ _
Used:
Hangman (10 life) - Input: a

Word: _ a _ _ _ a _
Used: a
Hangman (10 life) - Input: g

Word: _ a _ g _ a g _
Used: a g
Hangman (10 life) - Input: n

Word: _ a n g _ a g _
Used: a g n
Hangman (10 life) - Input: l

Word: l a n g _ a g _
Used: a g n l
Hangman (10 life) - Input: u

Word: l a n g u a g _
Used: a g n l u
Hangman (10 life) - Input: e

Word: l a n g u a g e
Used: a g n l u e
Hangman (10 life)

Congratulations!
The answer was language

```

## ■ 단어 선택

(1) 파일 dic.txt의 구성은 아래와 같다.

```

7
language
glasses
balloon
camera
football
puppet
newspaper

```

- 첫 번째 줄의 숫자는 전체 단어의 수를 나타낸다.
- 나머지 데이터들은 단어이며 줄(line)단위로 구분된다.
- 위의 파일 내용은 하나의 예제이며, 위와 같이 구성된 어떤 파일에 대해서도 처리할 수 있도록 구현해야 한다.

(2) 프로그램을 실행하면, 위의 파일에서 단어 1개를 랜덤하게 선택한다.

- 아래 예시를 참고하여 프로그램을 작성하라.

```

Let's play the hangman game!
Word: _ _ _ _ _ _ _
Used:
Hangman (10 life) - Input: a

Word: _ a _ _ _ a _ _
Used: a
Hangman (10 life) - Input: g

Word: _ a _ g _ a g _
Used: a g
Hangman (10 life) - Input: n

Word: _ a n g _ a g _
Used: a g n
Hangman (10 life) - Input: l

Word: l a n g _ a g _
Used: a g n l
Hangman (10 life) - Input: u

Word: l a n g u a g _
Used: a g n l u
Hangman (10 life) - Input: e

Word: l a n g u a g e
Used: a g n l u e
Hangman (10 life)

Congratulations!
The answer was language

Do you want to play again? (y/n) y

Word: _ _ _ _ _ _ _
Used:
Hangman (10 life) - Input: a

Word: _ _ a _ _ _ _
Used: a
Hangman (10 life) - Input: b

Word: _ _ a _ _ _ _
Used: a b
Hangman (9 life) - Input: c

```

⋮

```

Word: _ l a _ _ e _
Used: a b c d e f l q w y u z
Hangman (2 life) - Input: h

Word: _ l a _ _ e _
Used: a b c d e f l q w y u z h
Hangman (1 life) - Input: f
f: Already inserted character

Word: _ l a _ _ e _
Used: a b c d e f l q w y u z h
Hangman (1 life) - Input: p

Word: _ l a _ _ e _
Used: a b c d e f l q w y u z h p
Hangman (0 life)

Mission failed!
The answer was glasses

Do you want to play again? (y/n) n
Good bye!

```

## ■ 게임 종료

사용자가 선택된 단어를 life가 0이 되기 이전에 맞히거나, life가 0이 되면 게임이 종료된다. 게임이 끝나면, 게임을 다시 하겠느냐는 메시지가 출력되며 사용자가 n을 선택하면 프로그램이 종료된다. 사용자가 y를 선택하면 파일에서 랜덤하게 단어를 선택한 후 게임을 다시 시작한다. n을 선택하면 프로그램을 종료한다.

- 게임 재 시작 여부에 대한 사용자 입력은 y, n 이외의 입력에 대해서는 고려하지 않는다.

## ■ Problem 2: 삼목 게임 (40점)

### [문제]

- 2차원 배열을 이용하여 삼목 게임을 구현해 본다.

### [목적]

- 2차원 배열의 선언과 사용을 익힌다
- 함수에서 2차원 배열을 매개변수로 사용하는 방법을 익힌다.

### [주의사항]

- (1) 소스코드는 "assn2\_2.c" 로 저장할 것.
- (2) 보고서는 "assn2.docx", "assn2.hwp" 또는 "assn2.pdf"로 저장 한다. (보고서는 통합하여 작성)
- (3) 이 과제는 2차원 배열을 선언하고 사용하기 위한 과제이므로, 게임판은 반드시 2차원 배열로 선언 후 사용해야 한다. (마지막 페이지의 힌트를 참고할 것)
- (4) 전역 변수, goto 문, string 관련 함수, 구조체는 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능하다.
- (5) 모든 기능을 main 함수에서 구현한 경우 감점 처리한다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 함수를 정의해서 구현하여야 한다.
- (6) 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않는다.
- (7) 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 반드시 같게 작성해 주세요. (보드를 그리기 위해서 사용하는 문자는 -, + , | 로 통일, 공백의 개수는 자유, 불일치시 감점 있음)
- (8) 프로그램에서 랜덤 시드는 프로그램 시작 시 main() 에서 srand(time(NULL)); 함수를 한번만 호출하도록 하여 한번만 초기화 한다.

### [게임 규칙]

- (1) 보드의 크기는 3 x 3인 정사각형으로 모든 칸은 처음에는 비어 있다.
- (2) 게임에 참여하는 두 플레이어는 번갈아 가며 자기 차례에 하나씩 O나 X를 빈칸에 표시하여 차지한다.
- (3) 오직 빈칸에만 표시할 수 있다.
- (4) 먼저 가로, 세로 혹은 대각선 한 줄을 차지하는 플레이어가 이긴다.
- (5) 모든 빈 칸에 표시가 되었는데도 아무도 한 줄을 차지하지 못하면 비기는 경우가 된다.
- (6) 한 게임은 누군가가 이기거나 빈칸이 모두 채워질 때까지 진행된다.

### [설명 및 요구사항]

- (1) 삼목 게임을 컴퓨터와 대결하는 방식으로 구현하도록 한다.
- (2) 게임을 시작할 때 컴퓨터와 사용자가 중 돌을 누가 먼저 놓을지는 매 경기마다 임의로 결정되게 한다.

(3) 출력되는 보드의 좌표는 아래와 같다.

(0, 0)	(0, 1)	(0, 2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 3)

(4) 사용자는 자기 순서에, 보드의 좌표에 해당되는 2개의 정수 (정수 범위: 0~1 사이)를 입력한다. 틀린 범위의 값과 같은 잘못된 입력은 없다고 가정한다. 단, 사용자가 실수로 이미 채워진 값을 선택한 경우에는, 다시 좌표를 입력 받도록 한다.

(5) 컴퓨터는 자기 순서에, 임의로 놓을 곳을 정한다.

(6) 누군가가 가로, 세로, 혹은 대각선 한 줄을 차지하거나 더 이상 놓을 곳이 남아 있지 않은 경우 경기가 끝난다. 경기가 끝날 때 사용자가 이겼는지, 사용자가 졌는지 혹은 비겼는지에 대해 메시지를 출력한다. ('You Win!', 'Computer Wins!', 'Draw!' 중 출력)

(7) 출력되는 보드판은 빈칸을 '\*', 그 경기를 먼저 시작하는 쪽이 'O' (알파벳 대문자 O), 나중에 시작하는 쪽이 'X' (알파벳 대문자 X)로 표시한다.

(8) 한 경기가 끝나면 경기를 계속할지 묻는다. 사용자가 문자 y를 입력하면 새 경기를 시작하고 n을 입력하면 'Good bye!'를 출력하고 프로그램을 종료한다. 잘못된 입력은 고려하지 않는다.

(9) 이 문제를 해결하기 위해 반드시 아래 기능에 해당하는 사용자 정의 함수를 정의하고 사용해야 한다. 아래 기능 외의 필요한 함수를 정의해서 사용할 수 있다.

- **print\_board**: 보드판을 출력하는 기능
- **check\_board**: 게임의 승패를 검사하는 기능

### [실행 예시]

(1) 초기 실행 화면

- 컴퓨터와 사용자가 중 돌을 누가 먼저 놓을지 매 경기마다 임의로 결정되게 한다.

```
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
[User]
```

```
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
[Computer] 1 2
```

(2) 사용자가 먼저 돌을 놓는 경우, 빈 보드판이 출력되고 사용자의 입력을 기다린다.

```
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
[User]
```



- (3) 사용자가 (0, 0)의 자리에 돌을 놓았다. 사용자가 입력한 좌표가 반영된 보드가 출력되고 즉시, 컴퓨터는 (1, 1)의 자리에 돌을 놓았다.

```
[User] 0 0

+---+---+---+
| 0 | * | * |
+---+---+---+
| * | * | * |
+---+---+---+
| * | * | * |
+---+---+---+

[Computer] 1 1

+---+---+---+
| 0 | * | * |
+---+---+---+
| * | X | * |
+---+---+---+
| * | * | * |
+---+---+---+

[User]
```

- (4) 사용자가 이미 채워진 (1, 1)에 돌을 놓으려고 하면, 돌을 놓을 수 없는 자리이므로 다시 입력을 받아 (0, 2)에 돌을 놓았다. 컴퓨터는 (1, 0)에 돌을 놓았다.

```
[User] 1 1

[User] 0 2

+---+---+---+
| 0 | * | 0 |
+---+---+---+
| * | X | * |
+---+---+---+
| * | * | * |
+---+---+---+

[Computer] 1 0

+---+---+---+
| 0 | * | 0 |
+---+---+---+
| X | X | * |
+---+---+---+
| * | * | * |
+---+---+---+

[User]
```

- (5) 사용자가 (0, 1)의 자리에 돌을 놓았다. 사용자의 돌(O)이 먼저 대각선 한 줄을 채워 사용자가 게임을 이겼다. 게임을 계속 할지를 묻는 메뉴가 출력된다.

```
[User] 0 1

+---+---+---+
| 0 | 0 | 0 |
+---+---+---+
| X | X | * |
+---+---+---+
| * | * | * |
+---+---+---+

You Win!
Do you want to play again? (y/n)
```

컴퓨터가 이길 경우 아래와 같이 출력된다.

```
[Computer] 1 2

+---+---+---+
| 0 | * | 0 |
+---+---+---+
| X | X | X |
+---+---+---+
| * | 0 | * |
+---+---+---+

Computer Wins!
Do you want to play again? (y/n)
```

- (6) 아래는 사용자로부터 y를 입력 받아 새 경기를 시작한 경우의 예시이다.

```
[Computer] 1 2

+---+---+---+
| 0 | * | 0 |
+---+---+---+
| X | X | X |
+---+---+---+
| * | 0 | * |
+---+---+---+

Computer Wins!
Do you want to play again? (y/n) y

[Computer] 1 0

+---+---+---+
| * | * | * |
+---+---+---+
| X | * | * |
+---+---+---+
| * | * | * |
+---+---+---+

[User]
```

(7) 아래는 경기를 비기고 n을 입력 받아 프로그램을 종료한 경우의 예시이다.

```
+---+---+---+
| 0 | 0 | x |
+---+---+---+
| x | x | 0 |
+---+---+---+
| 0 | 0 | x |
+---+---+---+

Draw!
Do you want to play again? (y/n) n
Good bye!
```

## (힌트)

### ■ 배열 선언 및 사용

1. 배열의 크기를 선언할 때 변수를 사용하면 안됩니다(int array[i][i];). 충분한 크기로 선언한 뒤(int array[4][4];), 필요한 부분만 사용하면 됩니다.
2. 배열을 parameter 로 다른 함수에 넘겨주었을 경우, 그 함수에서 배열의 값을 바꾸면 main 함수에서도 바뀐 상태가 유지됩니다. 이는 배열이 포인터 형태를 띄고 있기 때문입니다.

```
#include <stdio.h>
void foo(int x[]);
int main()
{
    int a[10];
    a[0] = 0;
    foo(a);
    printf("%d\n", a[0]); //prints 1
    return 0;
}
void foo(int x[])
{
    x[0] = 1;
    return;
}
```

### ■ 어싸인 팁

1. 프로그램 작성 중 무한 루프에 의해 프로그램이 끝나지 않을 경우, Ctrl+C를 누르세요.