

Hangman in C

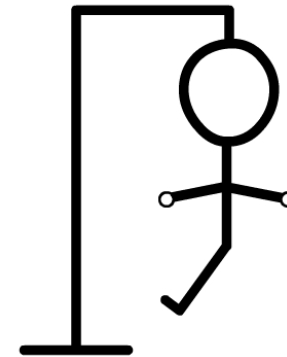
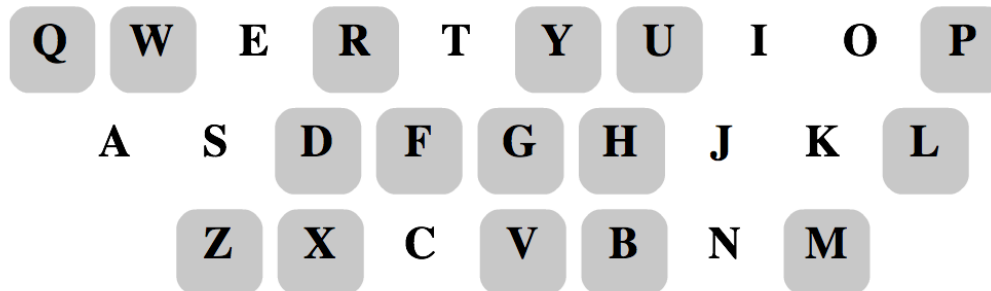
Sept 18

Hangman

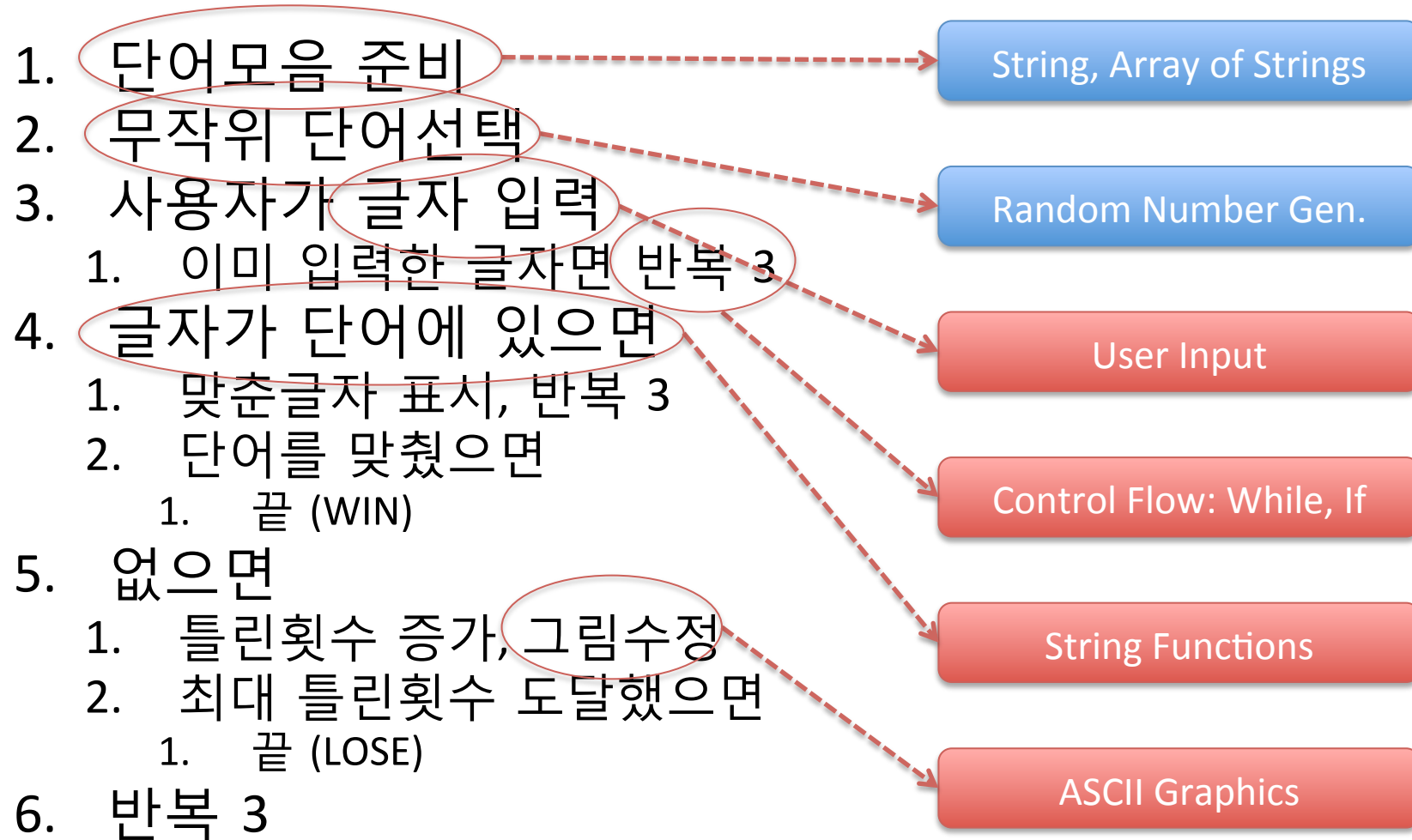
- 컴퓨터가 단어를 하나 생각하면 그 단어에 포함되어 있는 알파벳을 하나씩 추측해서 전체 단어를 맞추는 게임이다.
- 단어에 없는 알파벳을 추측할 때마다 벌점이 추가되어, 단어를 맞추기 전에 최대 벌점에 이르면 사람이 교수형에 매달리는 게임.

— — A — E

Click on the letters to guess which letters are in this word.
Make 8 wrong guesses and you lose.



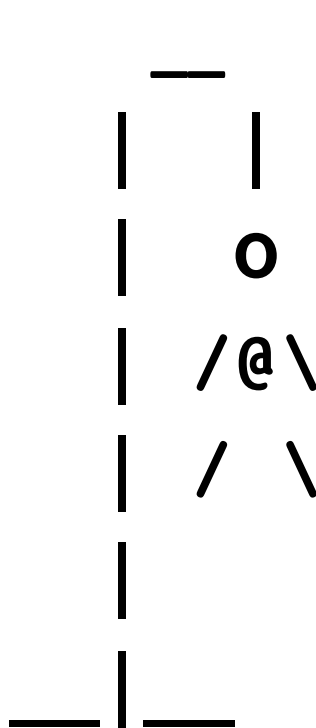
Program Flow



ASCII Graphics

[illegible][illegible]

Hangman ASCII Graphics

- 
- ```
 _
 | |
 | o
 | /@\\
 | / \\
 |
__|__
```
- printf로만 구현
  - 함수를 이용해서 코드 재사용
  - function draw\_hangman(int n)
  - draw\_hangman(0) : 빈 교수대
  - draw\_hangman(1) : 머리표시
  - draw\_hangman(2) : 머리+오른팔
  - draw\_hangman(6) : 교수형

# User Input

- `getchar()`
  - 한 글자만 입력
  - `c = getchar();`
- `scanf()`
  - 한 단어만 입력
  - `scanf("%s", &str);`

# 한줄 입력받기 (fgets)

- fgets( line, MAXLEN, stdin );
  - 사용자가 엔터를 칠 때까지(혹은 EOF), 단 최대 MAXLEN-1만큼 한 줄을 읽어서 (\n 포함) 변수에 저장하고 '\0'로 끝냄.
  - 입력을 저장할 배열의 용량을 표기함으로 안전.
- 문자열+'\n'을 최대 MAXLEN-1만큼만 쓰고 끝에 '\0'를 붙임
- `char line[6];`
- `fgets(line, 6, stdin);`
  - `hi↵` : `line={'h', 'i', '\n', '\0', ...}`
  - `hello↵` : `line={'h','e','l','l','o','\0'}`
  - `hello, world↵` : 상동

# String Functions

- **strcpy(s1, s2);**
  - Copies string s2 into string s1.
- **strcat(s1, s2);**
  - Concatenates string s2 onto the end of string s1.
- **strlen(s1);**
  - Returns the length of string s1.
- **strcmp(s1, s2);**
  - Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
- **strchr(s1, ch);**
  - Returns a pointer to the first occurrence of character ch in string s1.
- **strstr(s1, s2);**
  - Returns a pointer to the first occurrence of string s2 in string s1.



# Character Functions

- `int isalnum(int c);` if `c` is alphanumeric
- `int isalpha(int c);` if `c` is alphabetic only
- `int iscntrl(int c);` if `c` is a control character
- `int isdigit(int c);` if `c` is a numeric digit
- `int islower(int c);` if `c` is a lower case character.
- `int ispunct(int c);` if `c` is punctuation
- `int isspace(int c);` if `c` is space character
- `int isupper(int c);` if `c` is upper case character
- `int tolower(int c);` the corresponding lowercase letter
- `int toupper(int c);` the corresponding uppercase letter

# Game Loops

- 반복되는 상황은?
  - 사용자가 입력할 때마다
    - 다 맞추거나 다 틀릴 때까지 반복
    - Exit condition?
  - 입력한 글자가 알파벳이 아니면 반복 입력
    - Exit condition?
  - 입력한 글자가 이미 입력한 글자이면 반복
    - Exit condition?

# while vs do-while

- 모든 while문은 do-while로 대체 가능
- 모든 do-while문은 while로 대체 가능
- **Choose the simplest one**

<initialize>

```
while(<exit condition>) {
 <body>
}
```

-----

```
while(true) {
 <body>
 if(<exit condition>)
 break;
}
```

<initialize>

```
do {
 <body>
} while(!<exit condition>);
```

# Game Loop

```
repeat {
 print incorrect letters left, used letters
 repeat {
 repeat { get a letter
 } until it is alphabet
 } until it is new letter
 add to used letters
 if(hit) {
 update hit letters
 }
 else
 wrong count++
 print hangman
} until win or lost
```