

No. Checklist **제10회 10-3** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **void print (char \* string) printf ("%s\n", string);**

```

int main() {
    char * s = "C-Language";
    int i;
    for (i = strlen(s); i >= 0; i--) {
        printf ("%c", s[i]);
        s++;
    }
    return 0;
}

```

핵심내용: **strlen** 반환 값은 문자열의 길이

복수: **C-Language**

No. Checklist **제10회 10-4** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **bool ends\_with (char \* str, char \* pattern);**

```

#define CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#define true 1
#define false 0

bool ends_with (char * str, char * pattern) {
    int main() {
        char line [100] = "0";
        char suffix [5] = "0"; int count = 0;
        fgets (line, sizeof (line), stdin);
        fgets (suffix, sizeof (suffix), stdin);

        char * ptr = strtok (line, "\n");
        while (ptr != NULL) {
            if (ends_with (ptr, suffix) == true)
                count++;
            ptr = strtok (NULL, "\n");
        }
        return 0;
    }
}

bool ends_with (char * str, char * pattern) {
    int i = strlen (str);
    int j = strlen (pattern) - 1;
    if (i < j)
        return false;
    int start = i - j; int end = i;
    for (int k = start; k <= end; k++)
        if (str[k] != pattern[k - start])
            return false;
    return true;
}

```

핵심내용: **if (ends\_with (ptr, suffix) == true)**

복수: **Thinkthing**

No. Checklist **제10회 10-6** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **if (strcmp (gold [i], gold [j]) > 0) { temp = gold [i]; gold [i] = gold [j]; gold [j] = temp; }**

```

for (i = 0; i < size; i++)
    for (j = i + 1; j < size; j++)
        if (strcmp (gold [i], gold [j]) > 0) {
            temp = gold [i];
            gold [i] = gold [j];
            gold [j] = temp;
        }
for (i = 0; i < size; i++)
    printf ("%s", gold [i]);
printf ("\n");

```

핵심내용: **strcmp** 문자열 비교

복수: **Thinkthing**

No. Checklist **제10회 10-7** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **if (str[k] != pattern[k - start]) return false;**

```

bool ends_with (char * str, char * pattern) {
    int i = strlen (str);
    int j = strlen (pattern) - 1;
    if (i < j)
        return false;
    int start = i - j; int end = i;
    for (int k = start; k <= end; k++)
        if (str[k] != pattern[k - start])
            return false;
    return true;
}

```

핵심내용: **if (str[k] != pattern[k - start])**

복수: **Thinkthing**

No. Checklist **문제 10-2** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **int (\*matrix) [4] = num;**

```

int main() {
    int num [3][4];
    int m, n;
    int (*matrix) [4] = num;

    printf ("(m, n) 입력");
    scanf ("%d %d", &m, &n);
    printf ("value of m, n");
    printf ("%d", *matrix [m] [n]);
}

```

핵심내용: **\*matrix [m] [n]**

복수: **Thinkthing**

No. Checklist **문제 10-1** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **Char \* str;**

```

int doSomething (char * str) {
    str [0] = 'c';
    printf ("%s\n", str);
}

char myString [6];
doSomething (myString);

```

핵심내용: **Char \* str**

복수: **Thinkthing**

No. Checklist **문제 10-3** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **if (target < a [j]) a [j+1] = a [j];**

```

for (i = 1; i < size; i++) {
    target = a [i];
    for (j = i - 1; j >= 0; j--) {
        if (target < a [j])
            a [j+1] = a [j];
        else
            break;
    }
    a [j+1] = target;
}

```

핵심내용: **if (target < a [j])**

복수: **Thinkthing**

No. Checklist **문제 10-4** 중요도 ☆☆☆☆☆

과목: **C**

출처: **이것이 C는 아니다**

점수: **100**

메모: **if (a [i] > a [j]) { temp = a [i]; a [i] = a [j]; a [j] = temp; }**

```

for (i = 0; i < size; i++)
    for (j = i + 1; j < size; j++)
        if (a [i] > a [j]) {
            temp = a [i];
            a [i] = a [j];
            a [j] = temp;
        }

```

핵심내용: **if (a [i] > a [j])**

복수: **Thinkthing**



① 동적메모리 할당

\*include <stdlib.h>

void \* malloc (int size) → 크기라오

void \* calloc (int nitems, int size) → 크기라 X

void free (void \*ptr)

Ex) int \*p = (int\*) malloc (n \* sizeof(int));

free(p); ← 메모리를 풀이주는 코드  
(용자의 할당받은 기억장소가 해제되는 상황)

calloc은 0으로 채워주지만, malloc은 아니다.

→ 할당량이 초과하면, NULL이 반환

② String = array = pointer

char \*p = "abcde\0";

③ string.h

· strlen

1 strcmp (char \*str1, char \*str2)

같으면 0

str1 먼저 than str2 ⊖

str2 먼저 ⊕

ASCII

→ 대문자가 소문자보다 먼저!!

Ex) strcmp ("abcd", "aCd")

↓ 먼저 x 먼저 0

result = 1

1 strcpy (char \*dst, char \*src)

→ buffer overflow가 일어나지 않도록

destination의 공간을 충분히 만들어야 한다.

· strcpy (dst, src, n) ← element의 개수 (처음부터 n까지)

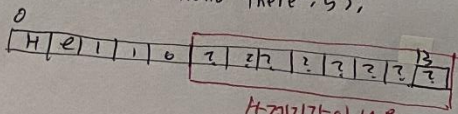
Ex) char str2[5];

strcpy (str2, "hello, world");

10은 카피 X

char str2[14];

strcpy (str2, "hello there", 5);



→ 레지스터의 사용

· strcat (char \*dst, char \*src);

↳ dst + src

strncat (dst, src, n);

↳ dst + src (n바이트)

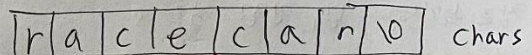
④ Substring

char chars[8];

strcpy (chars, "racecar");

char \*str1 = chars;

char \*str2 = chars + 4;



⑤ void convert (int years[]) {

int \* years

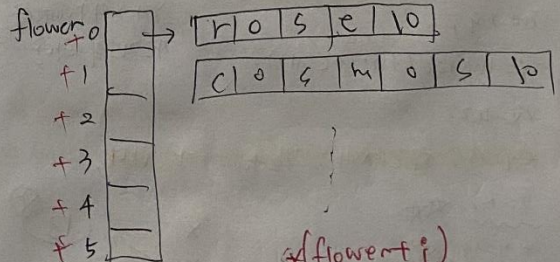
↓  
포인터 변경 처리...

⑥ str1 = "인어" (X)

\*str2 = "인어" (O)

문자열 = 배열!!

⑦ char \* flower[5] = { " ", " ", " ", " ", " " }



\*(flower + i)

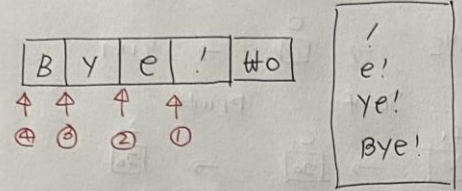


10 연습문제

```
8) note[] = "see you at the snack bar!"
ptr++ → e you at the snack bar!
```

```
note[7] = '!'
note[6] = 'e' → see you! 중략 X
```

```
9) Char *ptr, word[] = "Bye!";
ptr = word + strlen(word); word++
while (--ptr >= word) word+3 (--);
puts(ptr);
```



III Call by Value 미시는,

입력과 매개변수가 서로 다른 기억장소를 사용하며, 함수사용이 끝나면 매개변수의 기억장소는 사라진다. division 함수 실행이 끝나면 x, y, g, r의 값이 사라지며, main함수에서 g, r의 값은 초기화되지 않음으로, 쓰레기값이 출력된다.

Week 3) 구조체

1) 구조체

: 사용자 정의 Data type  
(다른 types를 하나의 group으로 가능)  
≡ 멤버

```
struct Tag {
```

Type Member 1; → 이러한 메모리도 할당되지 않음으로, 변수를 생성해야 함.  
Type Member 2;

3) 반드시 !!  
① 변수

- ① 처음부터 선언
- ② 별도의 line

```
2) struct Tag; 변수;
```

2) 구조체 멤버

- 멤버가 구조체일 수 있음!
- 동일 이름 같은 구조체 내 불가  
↳ 다른 " 가능

⇒ 멤버 연산자 '.' 사용해서 참조 가능.

```
Ex) struct point {
    int x;
    int y;
};
struct point pt = {10, 20};
int x = pt.x;
int y = pt.y;
```

*Nested Structures!*

```
Ex) struct rectangle {
    struct point t1;
    struct point t2;
};
```

```
struct rectangle rect;
int t1x = rect.t1.x;
int t1y = rect.t1.y;
```

```
Ex) printf("선수이름?");
scanf("%s", player.name);
```

char형 배열  
⇒ @사용하지 X

```
printf("1, 2, 3 라운드 점수?");
scanf("%d %d %d", &player.1, &player.2, &player.3);
```

int형

3) 구조체와 포인터

```
Ex) struct point {
    int x; int y;
};
struct point p = {10, 20};
struct point *pp = &p;
pp->x = 10; ○ pp.x = 10
pp->y = pp->y;
○ y = pp.y
```