

15주차 실습

2017-12-14

마지막 수업



과제 점수 확인

- 홈페이지에 과제 점수 파일을 업로드
- 과제 클레임은 12/17(일) 23:59 KST 까지
조교 메일(snupp2017@gmail.com)로 신청

기말고사 공지

- 12/17(일) 16:00 ~ 21:00

공지 및 환경 세팅을 위해 **15:40까지** 입실하기 바랍니다.

기말고사 공지

- 시험장소 : 소프트웨어 실습실 / 하드웨어 실습실
(302-311-1) (302-310-2)

시험장 및 좌석은 시험 당일 안내 예정

- 일요일에는 302동 1층 정문이 열리지 않습니다.
302동 2층 주차장 측의 문을 이용해서 건물에 들어오시면 됩니다.

기말고사 공지

- 시험 중간에는 채점 사이트를 제외한 외부 인터넷 망에 연결 X
- 문제 설명에 제시된 기본 데이터의 채점결과만 알 수 있음
- 에러 관련 질문은 받지 않습니다

자주 하는 실수들

- 지역변수 선언 후 초기화 X

```
int cnt;
while(cnt < 100){
    // do something
    cnt++;
}
```

- 지역변수로 큰 배열 잡기

```
int main(int argc, char* argv[]){
    int arr[10000000]; // stack overflow
}

void merge(...){
    int arr[500000];
    merge(...)
}
```

자주 하는 실수들

- String 끝에는 NULL이 와야

```
// char buf[10];  
char buf[11];  
scanf("%s", buf);    // "1234567890"
```

- char → 'c'
 string → "string"

```
char c;  
c = 'a';    // 0  
c = "a";    // X
```

```
char str[20] = 'week 15'    // X  
char buf[20] = "week 15"    // 0
```


자주 하는 실수들

- 잘못된 메모리 참조
* ~Signal 11 = Seg fault

```
int* ip;  
ip[3] = 500;    // segmentation fault
```

```
int* ip = (int*)malloc(sizeof(int) * 10);  
// or int ip[10];  
ip[3] = 500;    // 0  
ip[100] = 500;  // seg fault
```

```
// error if ptr == NULL  
ptr->next = ptr;
```

```
char* s;  
scanf("%s", s); // error!
```

자주 하는 실수들

- off-by-one

```
int arr[n];  
for(int i = 0; i < n - 1; i++)  
    sum += arr[i];
```

- 빼야 하는데 더하기

```
for(int j = n - 1; j >= 0; j++)
```

- 잘못된 인덱스 참조

```
for(int j = 0; j < n; i++)
```

자주 하는 실수들

- == 대신 = 쓰기

```
if(x = 2) { ... } // X  
if(x == 2) { ... } // O
```

- overflow

```
int x = 1e9, y = 1e8;  
int i = x * y; // 1569325056
```

- type casting 실수

```
long long l = x * y; // 1569325056  
long long l = (long long)x * y;
```

자주 하는 실수들

- 재귀 무한루프
(stack overflow)
- 함수 return이 없음
(undefined behavior)

```
int fib(int n){  
    return fib(n - 1) + fib(n - 2);  
}
```

```
int fib(int n){  
    if(n <= 1) return 1;  
    fib(n - 1) + fib(n - 2);  
}
```

자주 하는 실수들

- 괄호로 감싸지 않음
(변수의 스코프 문제)
- 잘못된 세미콜론의 사용
- 포인터 변수 선언 실수

```
for(int i = 0; i < n; i++)  
    for(int j = i + 1; j < n; j++)  
        if(a[i] > a[j])  
            tmp = a[i];  
            a[i] = a[j];    // Error!  
            a[j] = tmp;    // Scope is limited
```

```
int a = 0, i = 0;  
for(i = 1; i <= 100; i++);  
    a += i;    // a = 101
```

```
int *a, b;    // a is a pointer, but b isn't
```

자주 하진 않...는 실수들

- 같은 동작을 하는 코드

```
if(c == '0' || c == '1' || c == '2' ||  
    c == '3' || c == '4' || c == '5' ||  
    c == '6' || c == '7' || c == '8' || c == '9')  
{  
    int val = c - 48;  
}
```

```
if('0' <= c && c <= '9')  
    int val = c - '0';
```

자주 하진 않...는 실수들

```
k=0;
while(tmp[m][k]){
    if(tmp[m][k]==','){tmp[m][k]='\0';}
    if(tmp[m][k]=='.'){tmp[m][k]='\0';}
    if(tmp[m][k]=='!'){tmp[m][k]='\0';}
    if(tmp[m][k]=='?'){tmp[m][k]='\0';}
    if(tmp[m][k]=='*'){tmp[m][k]='\0';}
    if(tmp[m][k]=='@'){tmp[m][k]='\0';}
    if(tmp[m][k]=='#'){tmp[m][k]='\0';}
    if(tmp[m][k]=='$'){tmp[m][k]='\0';}
    if(tmp[m][k]=='^'){tmp[m][k]='\0';}
    if(tmp[m][k]=='&'){tmp[m][k]='\0';}
    if(tmp[m][k]=='('){tmp[m][k]='\0';}
    if(tmp[m][k]==')'){tmp[m][k]='\0';}
    if(tmp[m][k]=='['){tmp[m][k]='\0';}
    if(tmp[m][k]==']'){tmp[m][k]='\0';}
    if(tmp[m][k]=='{'){tmp[m][k]='\0';}
    if(tmp[m][k]=='}'){tmp[m][k]='\0';}
    if(tmp[m][k]=='~'){tmp[m][k]='\0';}
    if(tmp[m][k]=='~'){tmp[m][k]='\0';}
    if(tmp[m][k]=='/'){tmp[m][k]='\0';}
    if(tmp[m][k]==':'){tmp[m][k]='\0';}
    if(tmp[m][k]==';'){tmp[m][k]='\0';}
    if(tmp[m][k]=='\\'){tmp[m][k]='\0';}
    if(tmp[m][k]=='%'){tmp[m][k]='\0';}
    k++;
}
```

```
k = 0;
while(tmp[m][k]){
    if(!isalpha(tmp[m][k])) // ctype.h
        tmp[m][k] = '\0';
    k++;
}
```

Man page

- strcmp의 return 값이 뭘 의미하더라?
fopen의 인자로 무엇을 넣어줘야 하지?
- man page on linux : C library function에 대한 reference 내장
사용법 : man 3 [function name]
- 시험 중에도 참고 가능 :: 함수를 통째로 외울 필요는 없음

Man page : Example

```
MEMCPY(3)                                Linux Programmer's Manual                                MEMCPY(3)

NAME
    memcpy - copy memory area

SYNOPSIS
    #include <string.h>

    void *memcpy(void *dest, const void *src, size_t n);

DESCRIPTION
    The memcpy() function copies n bytes from memory area src to
    memory area dest. The memory areas must not overlap. Use
    memmove(3) if the memory areas do overlap.

RETURN VALUE
    The memcpy() function returns a pointer to dest.

CONFORMING TO
    SVr4, 4.3BSD, C89, C99, POSIX.1-2001.

SEE ALSO
    bcopy(3), memccpy(3), memmove(3), mempcpy(3), strcpy(3),
    strncpy(3), wmemcpy(3)

COLOPHON
    This page is part of release 3.54 of the Linux man-pages
    project. A description of the project, and information about
    reporting bugs, can be found at
    http://www.kernel.org/doc/man-pages/.
```

Task 15_1 Bingo!

Description

You will get 3 x 3 bingo table. For given number call, you have to check if the number call makes you bingo.

Input

3 x 3 bingo Table(9 integers)

number calls (integers). + number of calls will be lesser than 100.

Output

print "Bingo!\n" if the number call makes bingo. If the number **have called before, ignore it.**

Task 15_1 Bingo!

Example

Input	output
1 2 3 4 5 6 7 8 9 1 2 3 4 7	Bingo! Bingo!
2 3 4 5 6 7 8 9 1 2 4 8 1 6 6	Bingo! Bingo!

마지막 수업

- 모두들 한 학기동안 수고하셨습니다 ☺
- 앞으로 들을 과목 대부분에는 프로그래밍에 대한 별도 지도가 없음
 - 이 과목을 통해 충분한 프로그래밍 기본기를 얻어갔기 바랍니다
- 마지막까지 유종의 미를 거둘 수 있기 바랍니다

마지막 수업

