

C교육 3주차

201702075 조수환





INDEX



001/ 2주차 과제 풀이

002/ 배열

003/ 문자열

004/ 함수

005/ 과제



C교육 3주차

2주차 과제 풀이

- 과제 1
- 과제 2
- 과제 3



2주차 과제 풀이

과제 1



A. 주어진 조건을 만족하도록 조건문을 설계한다.

이후 입력받은 수에 알맞은 비율을 곱하여 모두 더한 수를 출력한다

```
int main(void) {
    int salary, sum = 0;

    for(int i=0 ; i<3 ; i++) {
        scanf("%d", &salary);

        if(salary < 3000000) salary = salary * 0.9;
        else if(salary < 5000000) salary = salary * 0.8;
        else if(salary <= 10000000) salary = salary * 0.7;
        else salary = 0;

        sum = sum + salary;
    }

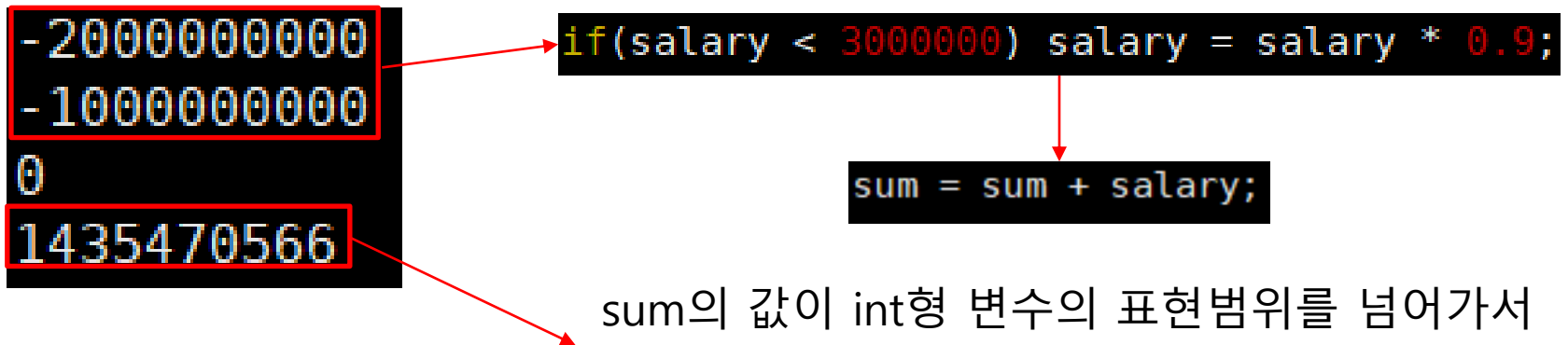
    if(sum >= 15000000) sum = sum * 0.9;

    printf("%d\n", sum);
    return 0;
}
```

2주차 과제 풀이 과제 2



- A. 음수를 표현할 수 있는 정수형 변수에 대하여, 음수에 대한 예외처리가 되어 있지 않으면 입력이나 연산 과정에서 자료형의 표현 범위를 넘어 오버 플로우가 발생 한다.



2주차 과제 풀이

과제 2



- A. 위와 같은 경우를 방지하려면 음수에 대한 예외 처리를 하거나 음수를 표현하지 않는 자료형을 사용한다. ex) unsigned int

음수에 대한 예외 처리, 음수를 입력받으면 0으로 초기화

```
if(salary < 0) salary = 0;  
else if(salary < 3000000) salary = salary * 0.9;  
else if(salary < 5000000) salary = salary * 0.8;  
else if(salary <= 10000000) salary = salary * 0.7;  
else salary = 0;
```

2주차 과제 풀이

과제 3



A. 공백을 출력하는 구간과, "*"을 출력하는 구간을 나눠서 반복문을 사용한다.

```
int main(void) {
    int n, start, end;
    scanf("%d", &n);
    for(int i=0 ; i<=2*n ; i++) {
        start = n-i, end = n+i;

        if(start < 0) start *= -1;
        if(end > 2*n) end = end + (2*(n-i));

        for(int j=0 ; j<start ; j++) printf("*");
        for(int k=start ; k<=end ; k++) printf(" ");
        for(int l=end ; l<2*n ; l++) printf("*");

        printf("\n");
    }
    printf("\n");
    return 0;
}
```

start, end : 공백 출력 구간 표시

공백 구간 이외는 "*" 출력

한 줄 출력이 끝나면 줄 바꿈

C교육 3주차

배열

- 배열과 메모리
- 실습



100명의 학생들의 시험점수를 입력받아서 저장하고자 한다.

```
int main(void) {  
    int student_1;  
    int student_2;  
    int student_3;  
    int student_4;  
    int student_5;  
    int student_6;  
    int student_7;
```

입력문(scanf) 상상도

```
scanf("%d %d %d %d %d %d...", student_1...);
```

student_... 이걸 100번을 한다고?



배열(Array)

- 같은 자료형의 변수들로 이루어진 유한집합
- 배열 선언 : <자료형> <배열 이름>[배열 크기] = <초기화>;
- ex) int student[5] = {1, 2, 3, 4, 5};
- 초기화는 배열 선언 이후에 따로 해도 됩니다.

```
int score[100];  
for(int i=0 ; i<100 ; i++) scanf("%d", &score[i]);
```

편-안



element, index

- 배열을 구성하는 각각의 데이터를 element(원소, 요소)라 한다.
- 각각의 element의 위치를 가리키는 숫자를 index라 한다.
- 배열의 인덱스는 0부터 시작한다.

```
int score[5] = {1, 2, 3, 4, 5};  
for(int i=0 ; i<5 ; i++) {  
    printf("score[%d] = %d\n", i, score[i]);  
}
```

```
sh2358@edu-argos:~/C-education/practice$ ./array  
score[0] = 1  
score[1] = 2  
score[2] = 3  
score[3] = 4  
score[4] = 5
```

만약 배열의 인덱스가 배열의 크기를 넘는다면?



반복문의 비교 식을 잘못 써서 인덱스가 score배열의 크기를 넘어버렸다.

```
int score[5] = {1, 2, 3, 4, 5};  
for(int i=0 ; i<=5 ; i++) {  
    printf("score[%d] = %d\n", i, score[i]);  
}
```

만약 자바에서 위의 코드 처럼 인덱스가 배열의 크기를 넘는다면

아마 아래와 같은 에러메시지를 출력 할 것이다.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
```



그런데 C에서는 요상한 숫자가 출력된다

```
sh2358@edu-argos:~/C-education/practice$ ./array
score[0] = 1
score[1] = 2
score[2] = 3
score[3] = 4
score[4] = 5
score[5] = 32766 ???
```

저걸로는 모자랐는지 인덱스에 음수를 집어넣어도

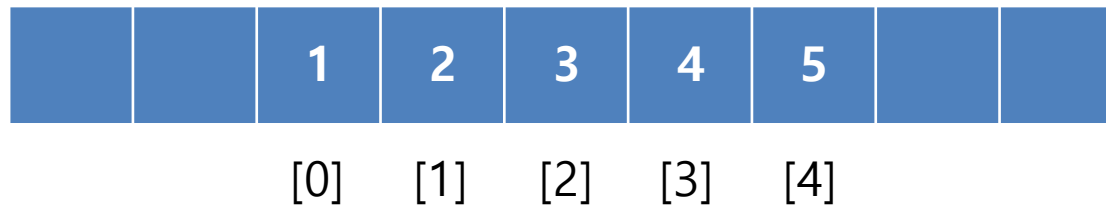
```
printf("score[-1] = %d\n", score[-1]);
```

```
score[-1] = 6
```

에러가 발생하지 않고 본 적 없는 숫자가 나온다. 이유가 뭘까?



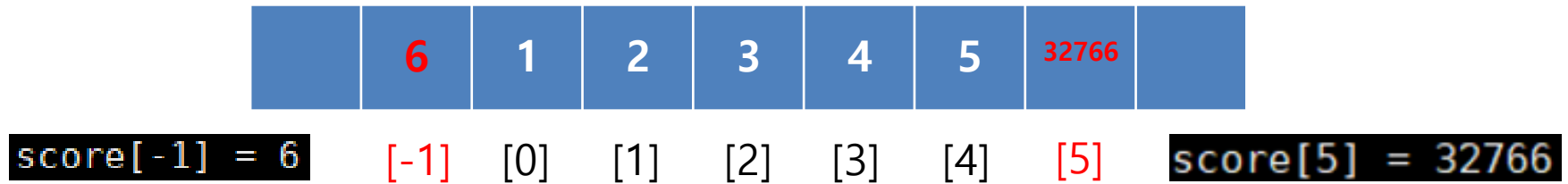
변수를 메모리에 저장하듯, 배열의 각 element 역시 메모리에 저장된다.



이전에 말했듯이 인덱스는 element의 위치를 가리키는 숫자이고,
C에서는 컴파일을 할 때 인덱스를 실제 메모리 주소로 변환한다.



그래서 인덱스가 배열의 크기를 넘어도 에러를 발생시키지 않는다.



위의 그림처럼 C에서의 배열 인덱스는 메모리 주소를 가리키기 때문에
가리키는 메모리 주소에 있는 값을 가져다 쓰는 것이다.

- * 현재 코드에서 사용하지 않은 메모리엔 이전에 해당 위치에서 사용한 값이 들어가 있다(쓰레기 값.) 이전 코드의 실행에서 출력된 숫자가 바로 이것.

배열 실습



8개의 숫자를 저장한 배열의 n번째 숫자와 m번째 숫자의 합을 출력하시오

```
#include <stdio.h>

int main(void) {
    int arr[8] = {3, 1, 76, 23, 12, 64, 32, 22};

    int n, m;
    scanf("%d %d", &n, &m);

    /*
        write your code
    */

    return 0;
}
```

3 5
88

-> $76 + 12 = 88$

1 3
79

-> $3 + 76 = 79$

7 8
54

-> $32 + 22 = 54$

C교육 3주차

문자열

- C언어의 문자열
- Buffer Overflow
- 실습





In Java : `String str = new String("hello!");`

`System.out.println(str);`

-> hello!

In C :

```
string str = "hello!";  
printf("%s\n", str);
```

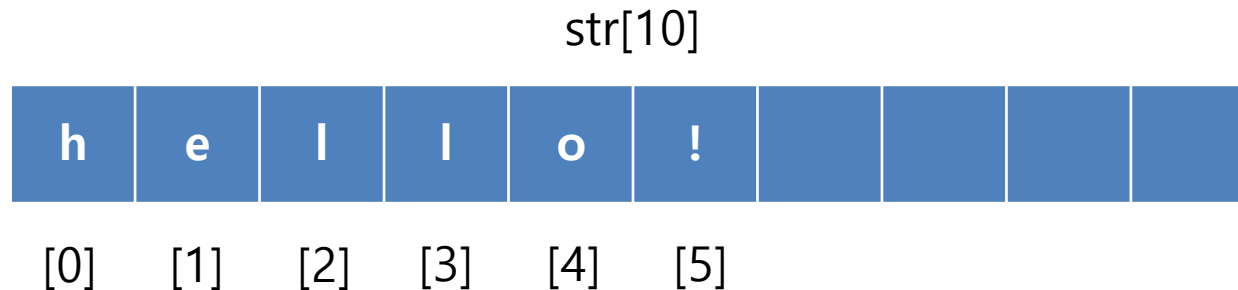
-> **error: unknown type name 'string';**

string이 뭔지 모르겠다고 함



C의 문자열

- 나열된 문자(char)들을 배열로 만들어서 문자열처럼 사용함.
- ex) `char str[10] = "hello!";`



```
char str[10] = "hello!";  
printf("%c\n", str[5]);
```

```
sh2358@edu-argos:~/C-education/practice$ ./string  
!
```



Null byte

- 문자열의 끝을 알리는 기능
- `\0`, `'\0'`, `NULL` 등으로 표현 가능하다.



Null byte가 없으면 어디까지가 문자열인지 모른다.



문자열 중간에 Null Byte를 넣은 코드

```
char str[100] = "Hello C World";  
str[7] = '\\0';  
printf("%s\\n", str);
```

```
sh2358@edu-argos:~/C-education/practice$ ./string  
Hello C
```

Null byte가 있는 str[7] 이전 까지만 출력한다.

아스키 코드(ASCII) – 숫자와 문자를 대응시키는 보편적 방식

10진	16진	문자	10진	16진	문자
64	0x40	@	96	0x60	`
65	0x41	A	97	0x61	a
66	0x42	B	98	0x62	b
67	0x43	C	99	0x63	c
68	0x44	D	100	0x64	d
69	0x45	E	101	0x65	e
70	0x46	F	102	0x66	f
71	0x47	G	103	0x67	g
72	0x48	H	104	0x68	h
73	0x49	I	105	0x69	i

```
char a = 65;
printf("%c\n", a);
return 0;
```

```
sh2358@edu-argos:~/C-education/practice$ ./ascii
A
```

전체 아스키 코드 표는 구글에 검색!



Buffer Overflow

- Buffer란 변수와 같이 데이터를 담을 수 있는 공간을 말한다.
- 말 그대로 Buffer의 크기를 넘어서 흘러 넘치는 현상.

```
char str[9];  
char score[100] = "000";  
scanf("%s", str);  
printf("Score : %s\n", score);
```

```
sh2358@edu-argos:~/C-education/practice$ ./string  
MyScore  
Score : 000
```

str[9]에 어떤 값을 입력하던 score[100]에는 "000"이란 값이 들어가 있으므로 printf는 당연히 000을 출력한다.



그런데 char 배열(str[9])의 크기보다 큰 문자열을 입력하면

```
sh2358@edu-argos:~/C-education/practice$ ./string
MyScoreIs999
Score : 999
```

갑자기 score[100]의 값이 999로 변한다.

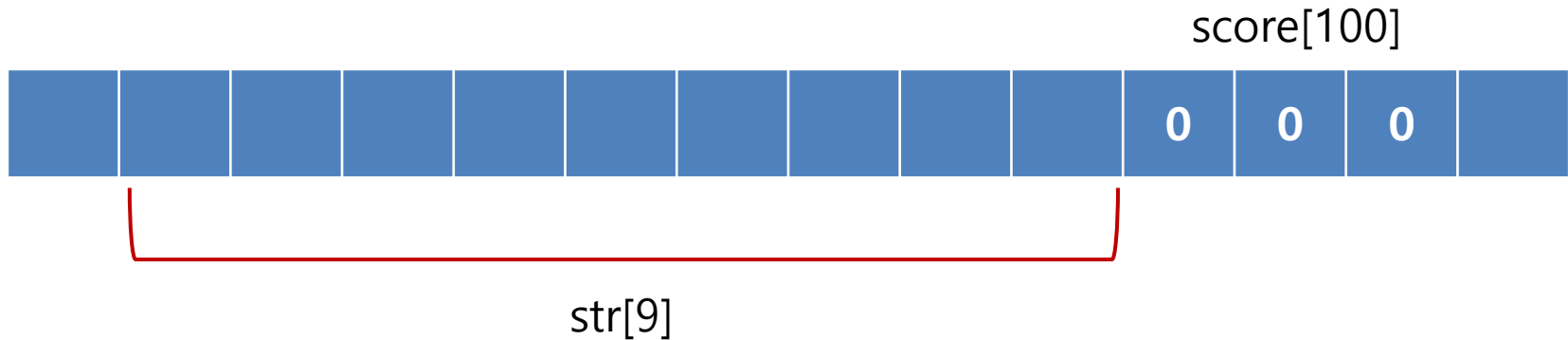
코드에는 score[100]의 값을 수정하는 부분이 전혀 없고,

입력받은 str[9]과 score[100]의 상호작용이 없는데도

어떻게 된 일일까?



문자열(str[9])을 입력받기 전, 메모리의 상태는 아래와 같다.



여기서 첫번째 입력(MyScore)을 하게 되면 아래와 같은 상태가 된다.





하지만 두번째 입력(MyScoreIs999)을 하게 되면

	M	y	S	c	o	r	e	I	s	9	9	9	₩0
--	---	---	---	---	---	---	---	---	---	---	---	---	----

위와 같은 상태가 되며, 여기서 score[100]을 printf로 출력하게 되면
printf는 시작 주소인 score[0](9)부터 Null byte까지를 문자열로 인식한다.

	M	y	S	c	o	r	e	I	s	9	9	9	₩0
--	---	---	---	---	---	---	---	---	---	---	---	---	----

score[100]

입력받은 문자열을 그대로 출력하는 프로그램을 작성하시오

```
sh2358@edu-argos:~/C-education/practice$ ./practice2  
hello!  
hello!
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice2  
bye  
bye
```

C교육 3주차

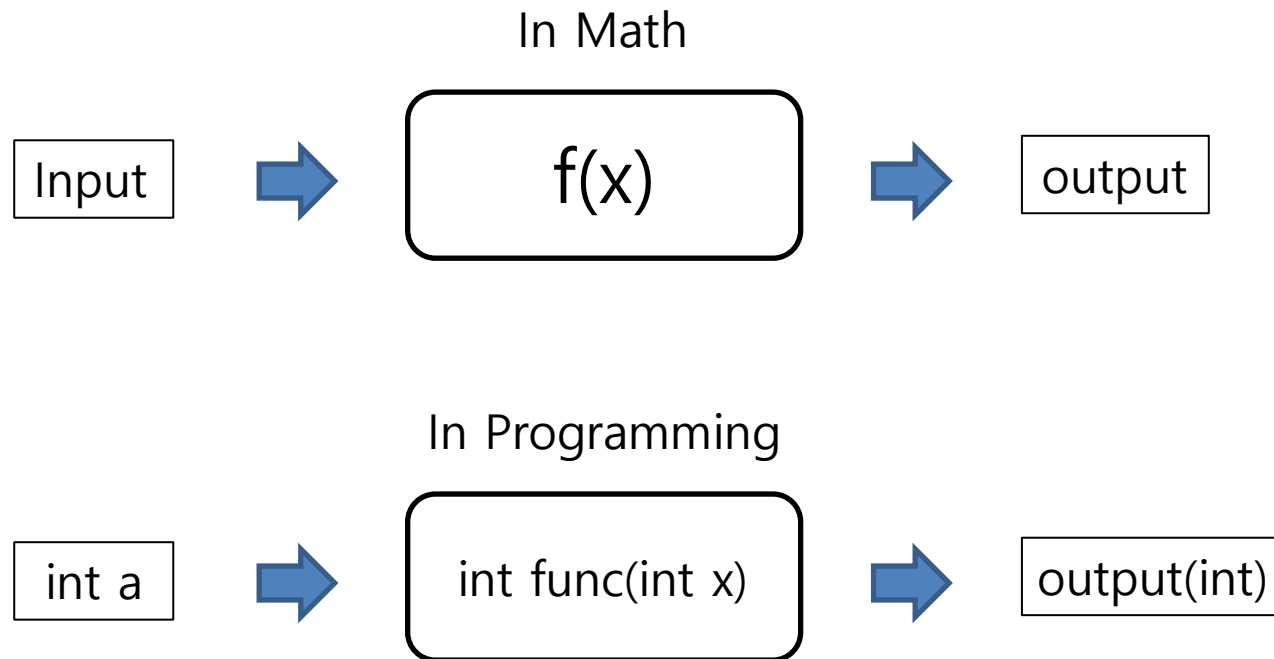
함수

- 함수의 개념
- 재귀 함수
- 실습





우리가 아는 수학적 함수와 별반 다르지 않다





함수(function)

- 특정한 목적의 작업을 수행하기 위해 설계된 프로그램 코드 집합
- Java에서 사용한 method와 비슷한 역할
- 선언 및 구현 : [반환값 자료형] [함수명]([인자]) { [코드 및 리턴값] }
- ex) int plus1(int x) { x++; return x }
- 변수를 선언할 때 처럼 함수명도 역할에 맞게 직관적으로!

```
int a = plus1(2);  
printf("%d\n", a);
```

```
sh2358@edu-argos:~/C-education/practice$ ./func  
3
```

$$f(x) = f(x - 1) + 2, f(0) = 0, (x \geq 0)$$

위의 점화식 처럼 자기 자신을 호출하는 함수를 재귀함수라 한다.

프로그래밍에서의 재귀 함수도 이와 같다.

```
int recursion(int x) {  
    if(x == 0) return 0;  
    else return recursion(x - 1) + 2;  
}
```

재귀 호출

참고로 위의 recursion 함수는 상단의 점화식을 정확히 동일하게 구현 한 것이다. Input에 따른 output도 같다.



$$f(x) = f(x - 1) + 2, f(0) = 0, (x \geq 0)$$

* 점화식 번역 *

$f(x) \rightarrow \text{recursion}(\text{int } x)$

$f(0) = 0 \rightarrow \text{if}(x == 0) \text{ return } 0;$

$f(x - 1) + 2 \rightarrow \text{recursion}(x - 1) + 2$

```
#include <stdio.h>

int recursion(int x) {
    if(x == 0) return 0;
    else return recursion(x - 1) + 2;
}

int main(void) {
    int x;
    scanf("%d", &x);
    printf("%d\n", recursion(x));
    return 0;
}
```

```
sh2358@edu-argos:~/C-education/practice$ ./rec
9
18
```

```
sh2358@edu-argos:~/C-education/practice$ ./rec
19
38
```


세 정수를 인자로 받고, 그 중 가장 큰 수를 반환하는 함수를 작성하시오.

```
#include <stdio.h>

int max_func(int a, int b, int c) {
    /*
        write your code
    */
}

int main(void) {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    printf("%d\n", max_func(a, b, c));
    return 0;
}
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice3
1 3 6
6
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice3
6 3 10
10
```

```
sh2358@edu-argos:~/C-education/practice$ ./practice3
38 29 40
40
```

C교육 3주차

과제

- 대문자 변환
- 피보나치 수열1(필수)
- 피보나치 수열2(도전)
- 회원 코드(도전)



대문자 변환(필수)



입력받은 소문자를 대문자로 바꿔 출력하시오.

(* 문자열을 입력할 때 공백(space)을 입력하시면 안됩니다.)

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework1
THIsIshomeworK1~~!!
THISISHOMEWORK1~~!!
```

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework1
maximumlengthis100
MAXIMUMLENGTHIS100
```

문자열의 길이는 최대 100입니다. 문자열 입력할때 공백을 입력하면

그 뒤로 출력이 안 됩니다. 이유는 다음주 과제풀이 시간에 알려드릴게요.

피보나치 수열1



피보나치 수열은 0과 1로 시작하며, 다음 항이 바로 앞의 두 항의 합인

수열을 말한다. (0, 1, 1, 2, 3, 5 ...)

1. 피보나치 수열의 n 번째 항을 출력하도록 fibonacci 함수를 작성

2. n 의 값에 각각 40, 45, 50을 입력하고 실행 시간을 비교 한 후,

왜 이런 결과가 나오는지에 대한 개인적인 생각 기술

(다음 슬라이드의 예제 코드를 따라 치고 함수만 구현하시면 됩니다.)



```
#include <stdio.h>
#include <time.h>

long long fibonacci(int n) {
    /*
        write your code
    */
}

int main(void) {
    int n;
    printf("n = ");
    scanf("%d", &n);
    clock_t start = clock();
    printf("f(n) = %lld\n", fibonacci(n));
    clock_t end = clock();
    printf("Run Time: %lf\n", (double)(end-start)/CLOCKS_PER_SEC);
    return 0;
}
```

Test Case

```
n = 10
f(n) = 55
Run Time: 0.000028
```

```
n = 20
f(n) = 6765
Run Time: 0.000249
```

```
n = 30
f(n) = 832040
Run Time: 0.009423
```

fibonacci 함수의 코드와 40, 45, 50의 입력에 대한 출력 결과,
실행시간의 차이에 대한 견해를 기술해서 보내주세요.



피보나치 수열의 n 번째 항을 구하는 프로그램을 작성하되

$n = 50$ 에 대한 실행 시간이 1초를 넘지 않도록 작성하시오.

(실행 시간을 구하는 코드는 아래 코드랑 이전 과제를 참고하세요)

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework3
n = 50
f(n) = 12586269025
RunTime : 0.000005
```



웹 사이트 운영자 철수는 사이트 회원들에게 회원 코드를 부여하고자 한다.

회원 코드에는 다음과 같은 조건이 있다.

- 회원 코드는 0으로 시작하지 않는다. (ex : 02467 (X), 01345643 (X))
- 인접한 숫자간의 차가 3을 넘으면 안된다. (ex : 1356 (O), 3012 (X))
- 모든 회원들에게 서로 다른 코드를 부여한다.

회원 코드의 자릿수를 n 이라 하면, n 자리의 회원코드로 몇 명의 회원을 받을 수 있는가? (ex : $n=4$, 1234, 1357, 2468, 4231, 4121)



($0 < n \leq 15$)라고 가정

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework4  
n = 4  
417
```

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework4  
n = 8  
72932
```

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework4  
n = 10  
965561
```

```
sh2358@edu-argos:~/C-education/homework/week3$ ./homework4  
n = 13  
46514780
```

상당히 어려우니까 안 풀린다고 실망하지 마세요.

Q & A

Thank You for Listening

