

Jaewon Choi
Keunsan Park
Haeun Lee
(snucsl.ta@gmail.com)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2023

4190.103A-001: Programming Practice Lab. 7



공지

공지

- 과제 late 제출
 - 실습과 동일하게, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
- 테스트 케이스는 공식적으로 공개 불가
- 이번주 Office hour는 오늘 수업 끝난 후 16:00 - 16:50

Lab. 6 실습 풀이

디버깅 실습 1

Q. 코드에 버그가 있습니다.

아래 문제를 읽고, 코드가 정상 동작하도록 디버깅을 해봅시다.

369게임 프로그램을 작성 해 봅시다.

프로그램은 자연수 n을 입력받은 후,

이어서 n개의 수를 입력 받습니다.

n개의 수를 받으면서 해당 수가 369게임 룰에 맞추어 입력되는지 확인합니다.

n개의 수가 모두 알맞으면 "Success",

틀린 입력이 들어오면 "Fail"을 입력하고 프로그램을 종료합니다.

```
for(int i = 1; i <= n; i++) // n 만큼 반복하면서,
{
    scanf("%d", &m); // m을 입력 받습니다.
    if(check(i) != m) // check 함수를 호출하여 입력이 올바른지 확인합니다.
    {
        printf("Fail"); // 값이 틀린 경우 Fail을 출력합니다.
        return 0;
    }
}
```

```
int temp = n; -> 원본값
while(n > 0) 보존
{
    if(n % 10 == 3 || n % 10 == 6 || n % 10 == 9)
        return 0;
    n /= 10; -> 6과 9에 대해서도 확인
}

return temp;
```

디버깅 실습 2

Q. 코드에 버그가 있습니다.

아래 문제를 읽고, 코드가 정상 동작하도록 디버깅을 해봅시다.

자연수 n 을 입력받은 후,

길이가 n 인 16진수를 입력 받습니다.

입력 받은 16진수를 10진수로 변환하여 출력합니다.

```
for (int i = 0; i < n; i++) // 16진수의 길이만큼 루프를 돌면 10진수로 변환합니다.
{
    if ( hex[i] >= 'A' && hex[i] <= 'F' )
        dec += hex[i] - 'A' + 10;
    else if ( hex[i] >= 'a' && hex[i] <= 'f' )    -> A ~ F = 10 ~ 15
        dec += hex[i] - 'a' + 10;
    else
        dec += hex[i] - '0';

    if(i == n-1)    -> 1의 자리는 곱할 필요
        break;    X
    dec *= 16;
}
```

실습 1

2개의 양의 정수(n, m) 를 입력 받아서,
 n degree 부터 $(n+m)$ degree 까지 1도 간격으로 \tan
값을 출력하는 프로그램을 작성 해 봅시다.

- 입력범위: n, m ($0 \leq n \leq 45, 1 \leq m \leq 3$)
- π 는 주어진 define 값을 사용하면 됩니다.
- 소수점 4번째 자리까지만 출력합니다.

```
int main()
{
    // Write your code here
    int n,m;
    scanf("%d %d",&n,&m);
    for(int i=0;i<=m;i++){
        printf("%.4f\n",tan((n+i) * PI / 180));
    }
    return 0;
}
```

실습 2

재귀 함수를 사용하여 아래와 같은 조화 수열의 n 번째 값을 구해봅시다. 입력으로 입력하는 수가 n 입니다. 배열을 사용하지 마세요.

$$\frac{2}{a_{n+1}} = \frac{1}{a_n} + \frac{1}{a_{n+2}}$$

위 조화 수열의 첫 두 항은 100, 80 입니다.

- $a(1) = 100, a(2) = 80$
- $n > 0$

```
double sequence(int n) {  
    // Write your code here  
    double sum;  
    if(n == 1)  
        return 100;  
    else if(n == 2)  
        return 80;  
    else{  
        sum = 2.0/sequence(n-1) - 1.0/sequence(n-2);  
        return 1.0/sum;  
    }  
}
```


실습 3

재귀 함수 `is_palindrome`을 작성하여 입력받은 문자열이 회문인지 아닌지 출력하세요. 회문이란 앞으로 읽어도 거꾸로 읽어도 똑같은 문자열을 말합니다.

예제 : 기러기, 우영우, 토마토

회문이면 1, 아니면 0을 출력하면 됩니다.

문자열 중간에 공백(space)는 skip 하세요. 입력받는 문자의 최대길이는 20입니다.

```
int is_palindrome(int left, int right) {
    // str[left,...,right]이 회문인지 판단하시오.
    //printf("l : %c , r : %c\n",str[left],str[right]);
    if (right - left < 1)
        return 1;

    if(str[left] == ' ')
        return is_palindrome(left + 1, right);
    else if(str[right] == ' ')
        return is_palindrome(left, right - 1);
    else if(str[left] != str[right])
        return 0;
    else
        return is_palindrome(left + 1, right - 1);

    // Write your code here
}
```

실습 4

1을 초항으로하고, 비가 3인 등비 수열은 아래와 같이 진행됩니다.

1, 3, 9, 27, ...

위 등비수열의 첫 번째 항부터 n번째 항까지 합을 구하는 프로그램을 재귀 함수를 이용하여 작성해 봅시다. (n의 최대 값은 41)

- 등비수열 값을 계산할 때, int 범위를 넘어 갈 수 있으니, unsigned long long int 사용을 권장합니다.

```
unsigned long long int func(int n) {  
    // Write your code here  
    long long int a = 1;  
    if(n == 1)  
        return a;  
  
    for(int i=1;i<n;i++)  
        a *= 3;  
    return a + func(n-1);  
}
```

Lab. 6 과제 풀이

과제 1

육면체 주사위의 경우의 수를 계산하는 프로그램을

작성해 봅시다.

자연수 n 과 m 을 입력받아서 주사위를 n 번 던져서 나온 값의 합이 m 이 나올 수 있는 모든 경우의 수를 세아리면 됩니다.

($2 \leq n \leq 7$)

($1 \leq m \leq 40$)

예를들어, $n = 2, m = 6$ 으로 입력이 주어지면,

가능한 경우는

(1,5) , (2,4) , (3,3) , (4,2) , (5,1)

로 총 5가지입니다.

```
int dice(int n, int m, int cnt, int sum)
{
    int ret = 0;
    if (cnt > n)
    {
        if (sum == m)
            return 1;
        return 0;
    }
    for(int i = 1; i <= 6; i++)
    {
        ret += dice(n, m, cnt+1, sum+i);
    }

    return ret;
}

int main()
{
    int n, m;
    scanf("%d %d", &n, &m);
    printf("%d", dice(n, m, 1, 0));
    return 0;
}
```

과제 2

두 자연수 n, m 을 입력 받은 후,
두 수의 중복된 수열의 최대 길이를 출력하는
프로그램을 작성 해 봅시다.
($0 < n, m < 2,147,483,647$)

예를들어,

n : 14356242

m : 43524

위와 같은 입력이 주어진 경우,

중복된 수열은

435,

24,

4,

3,

등이 있을 수 있습니다.

이 중에서 "435" 가 가장 긴 중복 수열이므로,

출력은 "3"이 됩니다.

```
int overlap(int n, int m, int l) {
    if (n == 0 || m == 0) return l;

    int a = 0;
    if (n % 10 == m % 10) {
        a = overlap(n / 10, m / 10, l + 1);
    }

    int b = overlap(n / 10, m, 0);
    if (l > b) b = l; // 이전 결과가 더 좋은 경우
    int c = overlap(n, m / 10, 0);
    if (l > c) c = l;

    int p = a < b ? b : a;
    int q = p < c ? c : p;
    return q;
}

int main()
{
    int n, m;
    scanf("%d %d", &n, &m);
    printf("%d", overlap(n, m, 0));
    return 0;
}
```

배열 & 포인터

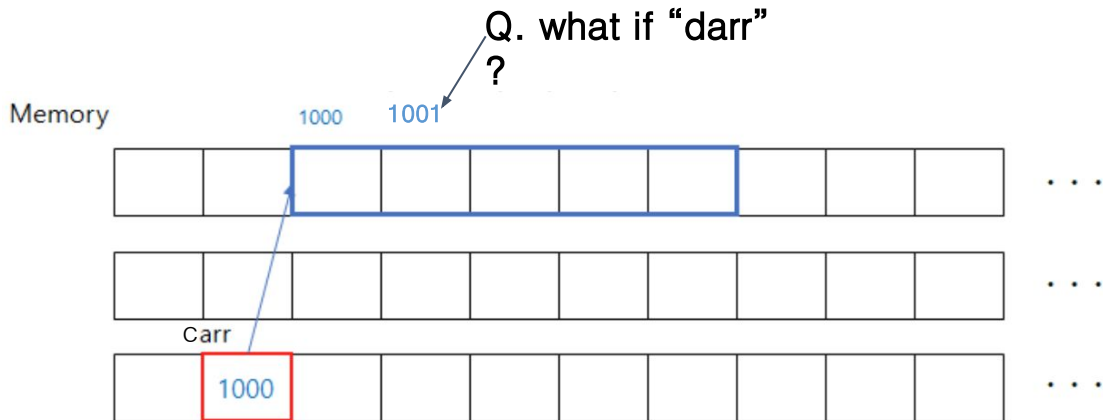
배열

- 배열 접근
 - `int age[3] = {0,};` // 0으로 배열 초기화.
 - `int i = 0;`
 - `age[i] = 20;` // index에 정수 형태 변수 넣는 것 가능.
 - `*(age + 1) = 21;` // age는 age[0]의 주소를 나타내는 포인터 변수이기도 함.
 - `age[i+3] = 23;` // 할당하지 않은 메모리를 접근 하기 때문에 문제 발생(e.g. segment fault).
주의필요

age	age + 1	age + 2
20	21	0

동적 할당

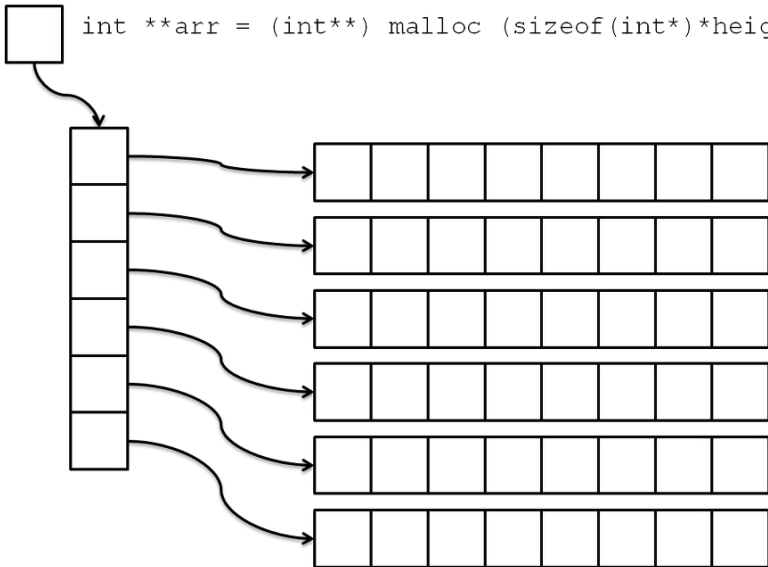
- 정적 할당
- `int age[3] = {0,};`
- 동적 할당
 - `char *carr; double *darr;`
 - `carr = (char*)malloc(sizeof(char)*5);`
 - `darr = (double*)malloc(sizeof(double)*5);`
 - `free(carr); free(darr); // 메모리 할당 해제 필수`



포인터 자료형 arr = (**포인터 자료형**)malloc(**자료형의 크기** * **원소의 개수**);

2차원 배열 동적 할당

```
int **arr = (int**) malloc (sizeof(int*)*height);
```



```
for(int i=0; i<width; i++){  
    arr[i] = (int*)malloc(sizeof(int)*width);  
}
```

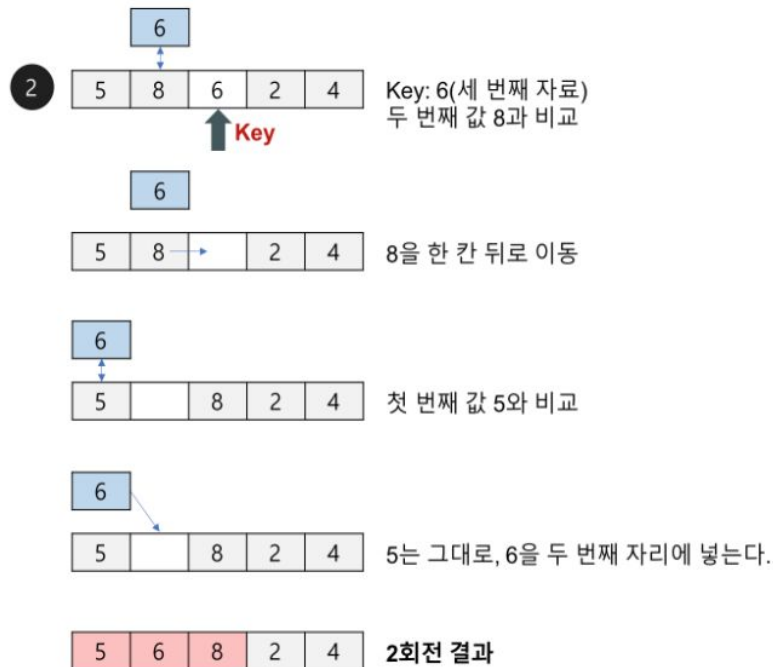
- $\&(\text{arr}[i][j]) == *(\text{arr} + i) + j$
- $\text{arr}[i][j] = *(*(\text{arr} + i) + j)$
- $\text{arr}[0][0] = **\text{arr} = *(*(\text{arr} + 0) + 0)$

정렬

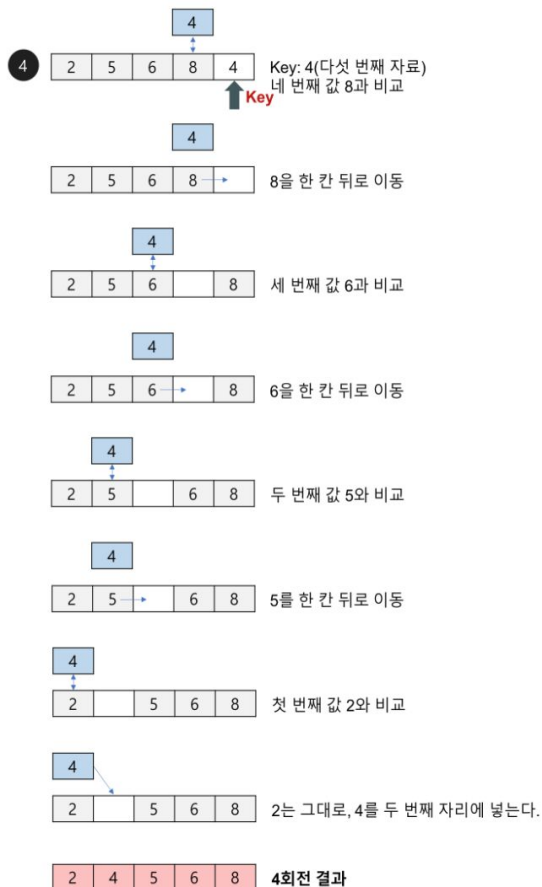
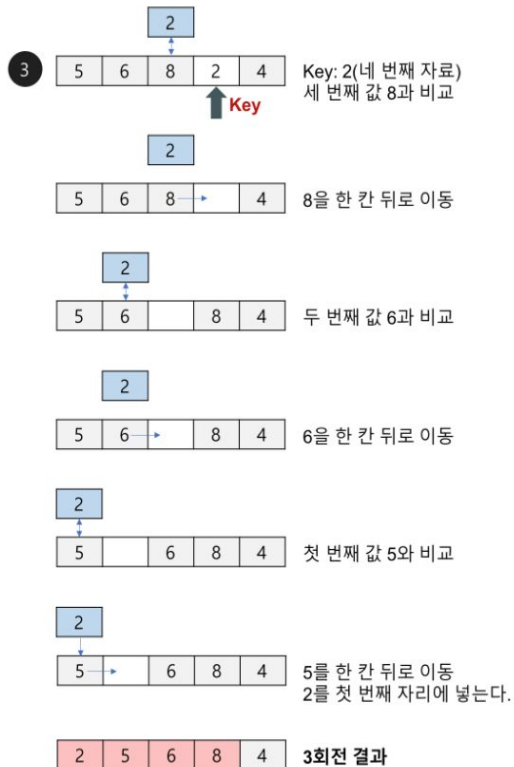
Insertion Sort

초기상태

8	5	6	2	4
---	---	---	---	---



Insertion Sort



실습 & 과제

실습 / 과제

- 과제 내용
 - 당일 진행한 실습과 관련된 문제
- 제출 기한
 - 실습 : 당일 자정까지, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
 - 과제 : 4/28(금) 23:59 까지, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
- 제출 방법
 - Elice의 Submit 기능 활용