



ALOHA

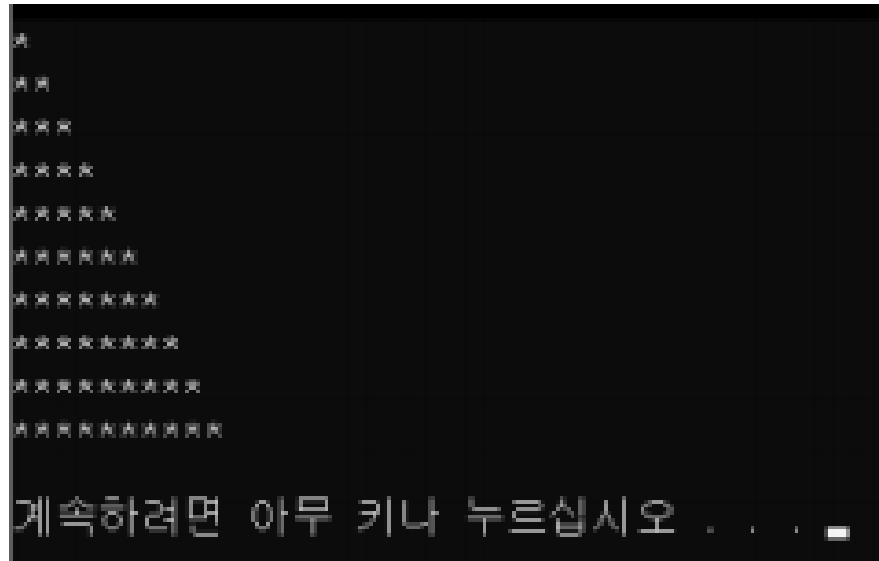
#3주차

반복문 응용 및 기본 DP

#CH.1

반복문 + 변수





반복문 + 변수

반복문으로 변수 i 의 크기를 1씩 증가시키면서 실행했어요.

지금까지는 반복문의 범위가 상수였어요.

반복문이 실행되는 횟수를 변수로 줄 수도 있어요.

반복문 + 변수

i를 1부터 10까지 반복

j를 1 부터 i까지 반복

```
for (int i = 1; i <= 10; i++){  
    for(int j = 1; j <= i; j++){  
        //do Something  
    }  
}
```

반복문 + 변수

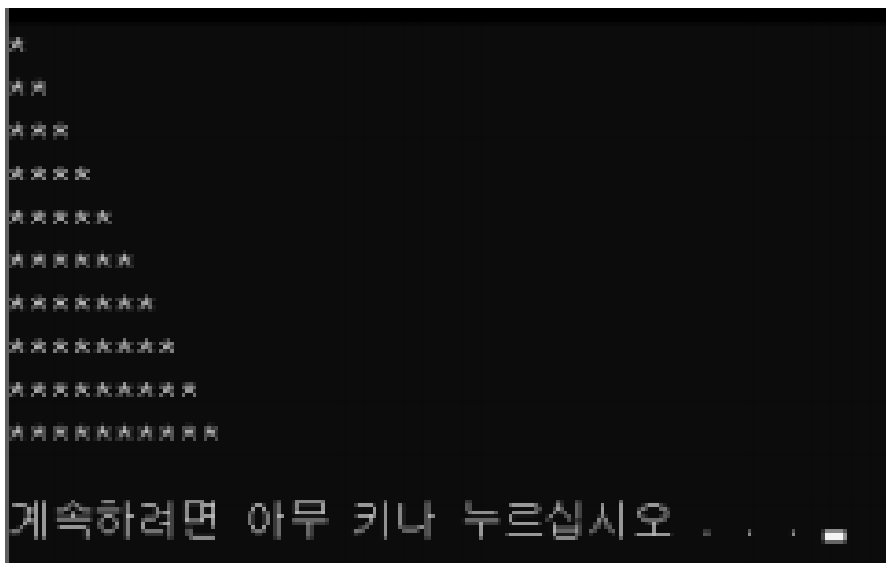
i를 a 부터 b까지 반복

```
for(int i = a; i <= b; i++){  
    //do Something  
}
```



이해 되셨나요??

목표



- 별 1개 출력하기
- 별 2개 출력하기
- 별 3개 출력하기
- ...
- 별 10개 출력하기



감이 잡힌 분들은 직접 짜보세요!

```
for( int i = 1; i <= 10; i++ ) {  
    별_i개_출력하기  
    한줄_띄우기  
}
```

별 찍기

어렵다고 겁먹을 필요 없어요!
천천히 보면서 같이 이해해 봅시다.

왼쪽 코드처럼 만 작성하면 되겠네요.

별을 i개 출력하는 방법만
따로 떼어 내서 생각해 봅시다.

```
for( int j = 1; j <= i; j++ ) {  
    printf("*");  
}
```

별 i개 출력하기

아까 배웠던 대로 1~ i 까지 반복하면서 *을 하나씩 출력하면 별이 총 i개 나오겠죠?

이제 앞부분이랑 다시 합쳐 볼 게요.

```
for( int i = 1; i <= 10; i++ ) {  
    for( int j = 1; j <= i; j++ ) {  
        printf("*");  
    }  
    printf("\n");  
}
```

별 찍기

어렵지 않죠?

#CH.2

반복문 + 배열



복습 - 배열

- 변수를 여러 개 만들기 귀찮을 때 써요.
- 배열이름[몇 번째 칸] 으로 변수처럼 쓸 수 있어요.
=> arr[2] 는 arr배열의 3번째 칸 (숫자는 0부터 셉니다)

예시 - 10개의 숫자 입력받기

- 아래처럼 하면 돼요.
- 인덱스(arr[i] 의 i) 는 0부터 세는 것에 주의합시다.
- arr[10]은 존재하지 않아요.

```
int arr[10];  
for( int i = 0; i < 10; i++ ) {  
    scanf( "%d", &arr[i] );  
}
```

실습 - 누적 합 구하기

- “누적 합 구하기” 라는 테크닉을 배울 거예요.
- 이 테크닉은 고급반에서도 써먹으니까 집중해서 해봐요.

#CH.2.1

누적 합 구하기





내용은 조금 어려울 수 있어요.

누적 합 구하기 - 문제

- N개의 숫자가 주어질 때,
- a번째 숫자부터 b번째 숫자까지 총 합을 구하고 싶어요.
- $[a, b]$ 를 모두 확인 하는 것보다 빠른 방법은 없을까요?

누적 합 구하기 - 아이디어

- $\sum_{i=a}^b arr_i = \sum_{i=0}^b arr_i - \sum_{i=0}^{a-1} arr_i$
- $[a, b]$ 의 합은 $[0, b]$ 의 합에서 $[0, a-1]$ 의 합을 뺀 것과 같아요.
- 예시) 3, 4, 5, 6번째의 합 = $[0, 6]$ 의 합 - $[0, 2]$ 의 합.

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7



A diagram showing a blue square followed by an equals sign, then a purple square followed by a minus sign, then a red square. This represents the equation: Blue = Purple - Red.



이해가 됐나요?

누적 합 구하기 - 구현

- 새로운 배열 `sum[N]` 을 만들 거예요.
- $Sum[i] = \sum_{k=0}^i arr[k]$
- 이렇게 하면 배열에 있는 두 값만 알면 어떤 구간의 합을 구할 수 있겠죠?

누적 합 구하기 - 해보기

한번 따라해 봅시다.

어떤 식으로 작동하는지 이해하는게 목표입니다.

언젠가 다시 배울 거니까 이해는 안돼도 괜찮아요.

```
int arr[15];
int sum[15];
for( int i = 1; i <= 10; i++ ) {
    scanf( "%d", &arr[i] );
}
for( int i = 1; i <= 10; i++ ) {
    sum[i] = sum[i-1] + arr[i];
}
```

누적 합 구하기 - 해보기

- arr 에는 입력되는 숫자가 들어간다.
- sum 에는 누적합이 들어간다.

```
int arr[15];
int sum[15];
for( int i = 1; i <= 10; i++ ) {
    scanf( "%d", &arr[i] );
}
for( int i = 1; i <= 10; i++ ) {
    sum[i] = sum[i-1] + arr[i];
}
```


#CH.3

기본 Dynamic Programming



DP란?

영어 그대로 해석하면, Dynamic programming = 동적 계획법 이에요.

다양한 알고리즘을 통해서,
시간이나 메모리 제한이 있는 복잡한 문제를,
작은 하위 문제들로 나누어서,
여러가지 기법을 더하여 문제를 해결해 나가는 과정 또는 방식을 의미한답니다.

수학문제를 풀 때처럼 일종의 설계를 하는 과정으로 이해하면 좋을 거 같아요.

DP란?

DP의 방식은 매우 다양해요.

대표적으로는 두가지로 나뉘는데

1. **Top-Down** 방식 : 큰 문제를 점점 작은 단위문제로 나누어서 푸는 방식
2. **Bottom-Up** 방식 : 작은 문제들을 해결하면서 점점 큰 문제에 접근하는 방식

우리는 이번주에 **Bottom-Up** 방식만 확인할게요 (Top-Down 방식에서 쓰이는 재귀함수가 다음주에 배우기 때문)

#CH.3.1

Fibonacci Sequence



Bottom-up 이란?

작은 문제를 해결하면서 점점 큰 문제에 접근하는 방식이에요.

예를 들어서, 7번째 피보나치 수를 구하기 위해서 $F(1)$, $F(2)$, \dots , $F(6)$ 를 순서대로 구해 나가는 과정이 있겠네요.

그리고 **한번 구해낸 결과값은**
매번 다시 계산하지 않고, 배열에 저장해서 다시 꺼내 쓰면 빠르겠네요!

Bottom-up 이란?

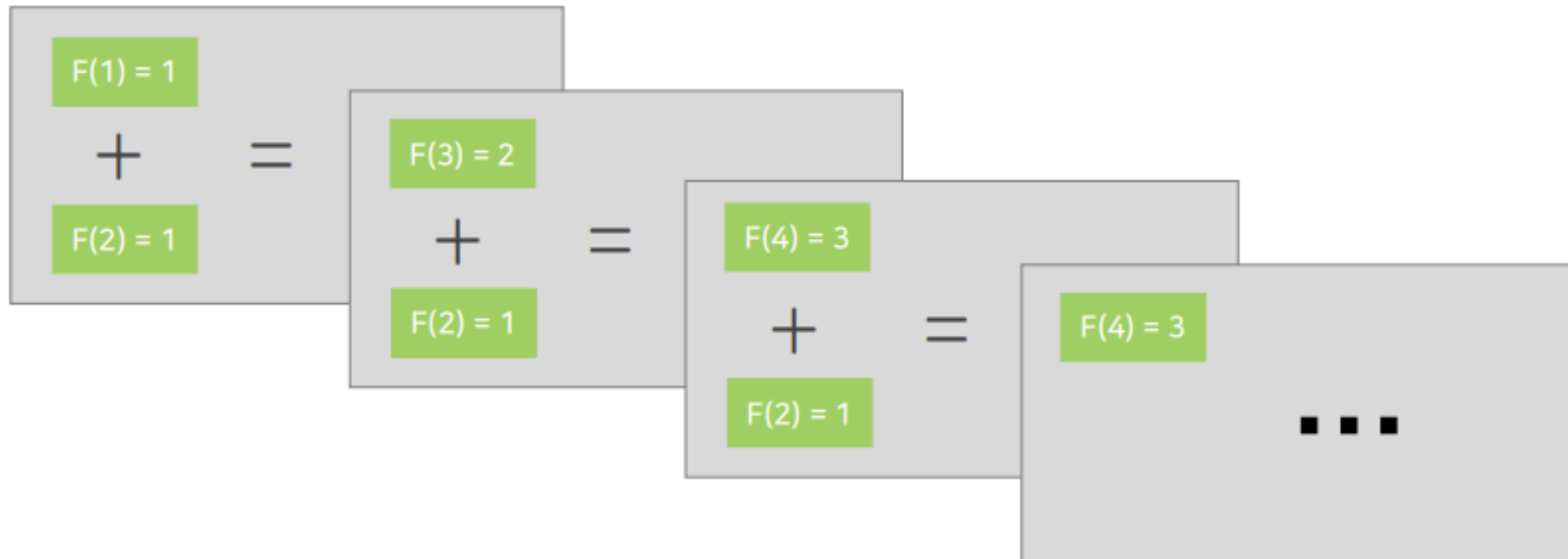
우리는 N번째 값에 도달하기 위해 **수열의 항들 사이의 관계**를 알아야 해요.

기본적으로 $F(1) = 1$, $F(2) = 1$ 인 것은 알고있는 상태에서 다른 항과의 관계는 어떨까요?

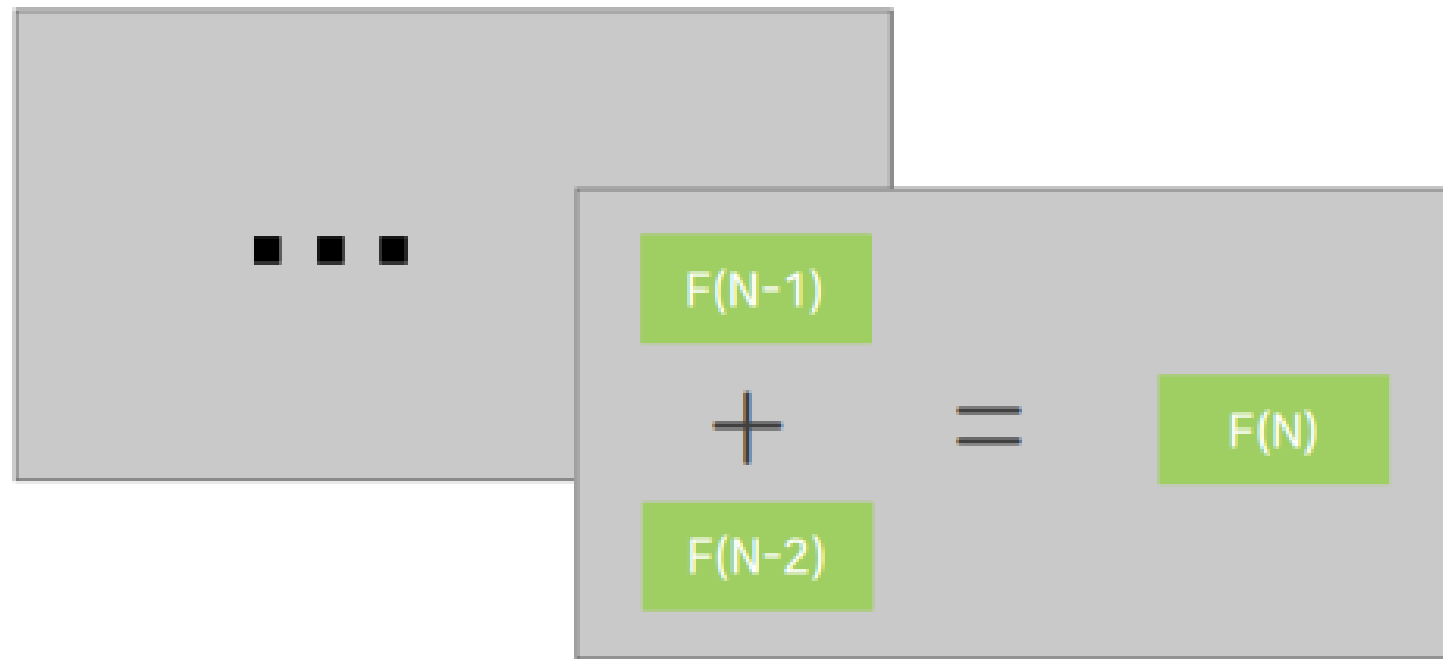
이걸 **점화식**이라고 해요. (피보나치 수열의 경우 $F(n) = F(n-1) + F(n-2)$)

이 점화식을 통해 값을 차근차근 구해 나가서 N까지 구하면 되겠죠?

Bottom-up 이란?



Bottom-up 이란?



그림으로 보니까 더 쉽죠?


```

1 | #include <stdio>
2 |
3 | int fib[10000];
4 |
5 | int main(void)
6 | {
7 |     int N;
8 |     scanf("%d", &N);
9 |
10 |    fib[1] = 1;
11 |    fib[2] = 1;
12 |
13 |    for (int i = 3; i <= N; i++)
14 |    {
15 |        fib[i] = fib[i - 1] + fib[i - 2];
16 |    }
17 |
18 |    printf("%d\n", fib[N]);
19 |
20 |    return 0;
21 | }

```

전역변수는 0으로 초기화 돼요

앞에서 말한 점화식이예요

N번째 피보나치 수

N 을 입력으로 받고
N번째 피보나치 수를 구하는 코드입니다.

직접 한번 해봐야 겠죠?

DP_Bottom-up

피보나치를 보면서 알게 된 사실이 하나 있죠?
우리가 DP 문제를 Bottom-up 방식으로 접근하려면

1. 초기값
2. 점화식

이 두가지를 알아야 해요. 매우 중요하니까 다양한 문제를 통해 연습해야 해요.

풀어볼까유



#2747
피보나치 수

DP 연습

풀어볼까유



#13301
타일 장식물

DP 연습

풀어볼까유



#2748

피보나치 수 2

DP 연습

풀어볼까유



#13699
점화식

조금 다르게 생긴
점화식

부록 - while문으로 별찍기

While 도 반복문이다.

```
int i, j;
i = 1;
while ( i <= 10 ) {
    j = 1;
    while( j <= i ) {
        printf( "*" );
        j++;
    }
    i++;
    printf( "\n" );
}
```

부록 - 수식으로 별 찍기

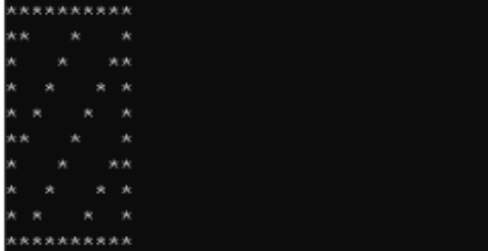
부등식이 나타내는 범위로 찍을 수 있다.

```
for( int i = 1; i <= 10; i++ ) {  
    for( int j = 1; j <= 10; j++ ) {  
        if( i >= j ) printf("*");  
    }  
    printf("\n");  
}
```


부록 - 수식으로 별찍기 응용

이런 것도 된다.

```
for( int i = 1; i <= 10; i++ ) {  
    for( int j = 1; j <= 10; j++ ) {  
        if( i == 1 || j == 1 || i == 10 || j == 10 || (i + j) % 4 == 0 )  
            printf("*");  
        else  
            printf(" ");  
    }  
    printf("\n");  
}
```



계속하려면 아무 키나 누르십시오 . . .



다음 시간에 만나요~