



ALOHA

#3주차멘토링

반복문 응용 및 기본 DP

Combination을 점화식으로 나타내기

고등학교 확통 시간에 배웠던 조합(combination)을 구현하는 방법?

$$C(n, r) = n! / (r! * (n-r)!)$$

이 식이 가장 쉽게 떠오르죠. 하지만 팩토리얼은 매우 비효율적이기 때문에 다른 방법을 찾아보면

$$C(n, r) = C(n-1, r) + C(n-1, r-1)!!$$

이 점화식을 이용할 수 있어요. ...점화식?! 오늘 배운 DP에 딱 알맞은 내용인걸?!

뭐라구요? 저거 파스칼삼각형으로 할 수 있는 거 아니냐구요? 똑똑하시네요!

이제 조합을 DP로 구현할 수 있게 되었어요!

파스칼 삼각형을 구현하면 조합을 구현한거나 마찬가지겠네요!

조합(Bottom-Up 활용)

nCr 을 $c[n][r]$ 로 나타내자

```
1  #include <stdio.h>
2
3  int c[100][100];
4
5  int main()
6  {
7      int n, r;
8      scanf("%d%d", &n, &r);
9      c[1][0] = 1;
10     c[1][1] = 1;
11     for ( int i = 2; i <= n; i++ )
12         for ( int j = 0; j <= i; j++ )
13             c[i][j] = c[i - 1][j - 1] + c[i - 1][j];
14     printf("%d\n", c[n][r]);
15     return 0;
16 }
```

파스칼 삼각형(Bottom-Up 활용)

```
1  #include <stdio.h>
2
3  int pascal[100][100];
4
5  int main()
6  {
7      int a, b;
8      scanf("%d%d", &a, &b);
9      pascal[1][0] = 1;
10     pascal[1][1] = 1;
11     for ( int n = 2; n <= a; n++ )
12         for ( int r = 0; r <= n; r++ )
13             pascal[n][r] = pascal[n - 1][r - 1] + pascal[n - 1][r];
14     printf("%d\n", pascal[a][b]);
15     return 0;
16 }
```

체크해보자!



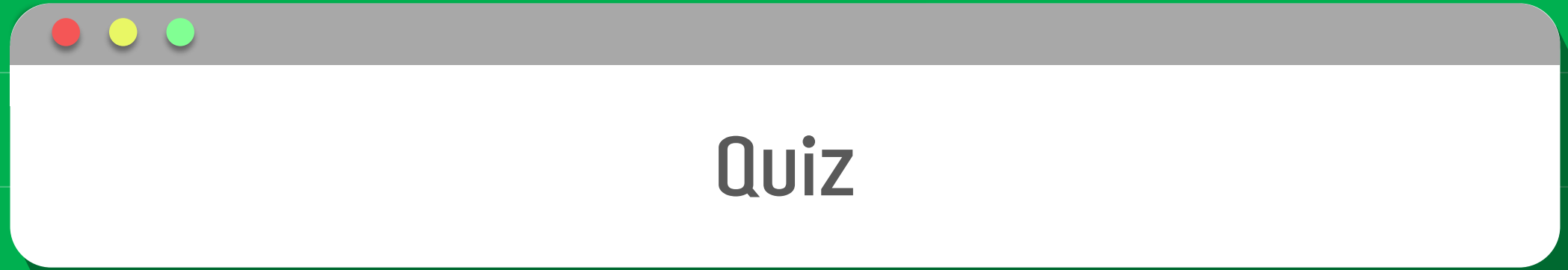
이중 for문과 이차원 배열 잘 다뤄보자



점화식과 초기값 꼼꼼히 설정하자



배열의 인덱스에 의미 부여할 수 있다



Q1

Q1. 다음처럼 출력되도록 코드를 짜시오. (그냥 프린트 금지)

1

2 3

4 5 6

7 8 9 10

Q2

Q2. 다음과 같은 파스칼의 삼각형을 점화식을 이용해 출력하시오. (그냥 프린트 금지)

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```




Answer

A1

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int sum = 0;
6
7      for (int i = 1; i <= 4; i++) {
8          for (int j = 1; j <= i; j++) {
9              printf("%d ", ++sum);
10             }
11             printf("\n");
12         }
13     }
```

A2

```
1  #include <stdio.h>
2
3  int pascal[10][10];
4
5  int main()
6  {
7      pascal[0][0] = 1;
8      pascal[1][0] = 1;
9      pascal[1][1] = 1;
10
11     for (int n = 2; n <= 5; n++) {
12         for (int r = 0; r <= n; r++) {
13             pascal[n][r] = pascal[n - 1][r - 1] + pascal[n - 1][r];
14         }
15     }
16
17     for (int n = 0; n <= 5; n++) {
18         for (int j = 1; j <= 5 - n; j++) {
19             printf(" ");
20         }
21         for (int r = 0; r <= n; r++) {
22             printf(" %d", pascal[n][r]);
23         }
24         printf("\n");
25     }
26 }
```

A2

혹시 13번 줄에 의문을 가진 사람이 있나요? 아닛, r이 0일 때 인덱스가 -1이 되잖아?! 라고 생각하셨다면 잘 발견하셨습니다! 그렇다면 왜 저 코드가 에러를 뱉지 않고 돌아가는 것일까요?

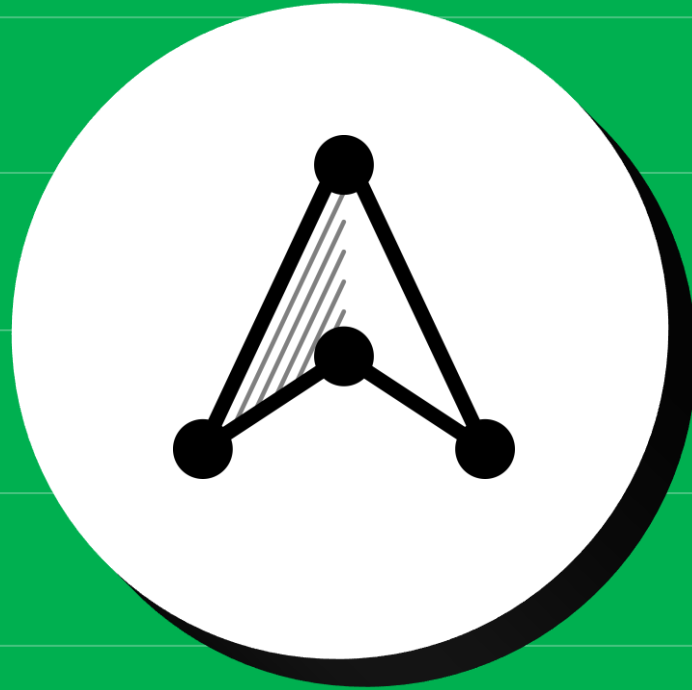
놀랍게도 `pascal[2][-1]`은 `pascal[1][9]`와 같습니다!

이차원배열은 사실 메모리상에서는 이차원이 아닌 일차원으로 쭉 이어져 있는데요, `pascal`은 10×10 배열이므로 `[1][9]` 다음에 `[2][0]`이 오게 됩니다.

따라서 `[2][0]`의 한 칸 이전 칸을 의미하는 `[2][-1]`은 `[1][9]`칸을 가리키게 되는 것이죠.

`pascal`배열은 전역변수로 선언하였기에 `pascal[1][9]`는 0으로 초기화되어 있고, 연산에 문제를 가져오지 않았기 때문에 올바른 값이 출력되는 것입니다.

썸썸하다면 `pascal[*][0]`을 1로 모두 초기화 하고 r은 1부터 계산을 하여도 무방합니다!



다음 시간에 만나요~