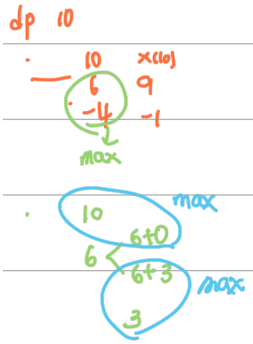
〈동적계획법〉

1. 연속합

10 -4 3 1 5 6 -35 12 21 -1




```python
n = int(input())
nums = list(map(int, input().split()))
dp = [[-1001, -1001] for i in range(n)]

dp[0] = nums[0]

if n >= 2:
  dp[1][0] = dp[0]
  dp[1][1] = max(dp[0] + nums[1], nums[1])

for i in range(2, n):
  dp[i][0] = max(dp[i-1][0], dp[i-1][1])
  dp[i][1] = max(dp[i-1][1] + nums[i], nums[i])

if n == 1:
  print(dp[0])
else:
  print(max(dp[-1]))
```

2. 평범한 배낭

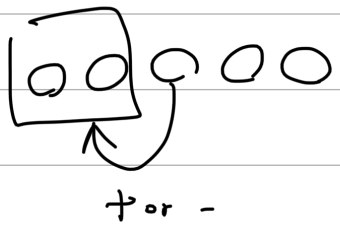
최대무게 7

6 13

4 8

3 6

5 12




```python
n,k=map(int, input().split())
memo=[[0]*(k+1) for _ in range(n+1)]

for i in range(1,n+1):
  w,v=map(int,input().split())
  for j in range(1,k+1):
    if j <w:
      memo[i][j]=memo[i-1][j]
    else:
      memo[i][j]=max(memo[i-1][j], memo[i-1][j-w]+v)
print(memo[n][k])
```

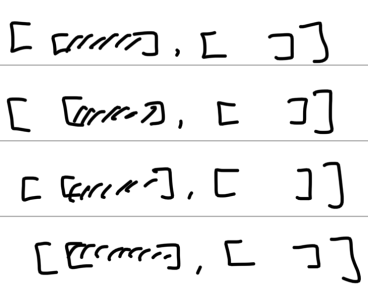
(4,7)

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

<분할정복>

1. 색종이 만들기

[[1 1 0 0 0 0 1 1],
[1 1 0 0 0 0 1 1],
[0 0 0 0 1 1 0 0],
[0 0 0 0 1 1 0 0],
[1 0 0 0 1 1 1 1],
[0 1 0 0 1 1 1 1],
[0 0 1 1 1 1 1 1],
[0 0 1 1 1 1 1 1]]



2. 쿼드트리

1 1 1 1 0 0 0 0 ((110(0101))(0010)1(0001))
1 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0
0 0 0 1 1 1 0 0
1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 0
1 1 1 1 0 0 1 1
1 1 1 1 0 0 1 1

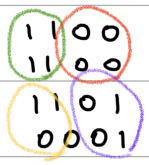
```python
n = int(input())
tree = [list(map(int,(input()))) for _ in range(n)]
result = []

def quad_tree(x,y,n):
    global result
    color = tree[x][y]

    for i in range(x, x+n):
        for j in range(y, y+n):
            if color != tree[i][j]:
                result.append("(")
                quad_tree(x,y,n//2)
                quad_tree(x, y+n//2, n//2)
                quad_tree(x+n//2, y, n//2)
                quad_tree(x+n//2, y+n//2, n//2)
                result.append(")")
                return
    result.append(color)

quad_tree(0,0,n)
print("".join(map(str,(result))))
```

→ 색깔이 다른 경우

1100 [[1100],
1100 [1100],
1101 [1101],
0001 [0001]]

result

["c"]



["c", 1]

["c", 1, 0]

["c", 1, 0, "c"]

["c", 1, 0, "c", 1, 1, 0, 0, "c"]

["c", 1, 0, "c", 1, 1, 0, 0, "c", "c", 0, 1, 0, 1, "c", "c"]

3. 종이의 개수

```python
import sys
input = sys.stdin.readline

n = int(input())
minus_cnt, zero_cnt, plus_cnt = 0, 0, 0

papers = []
for _ in range(n):
    row = list(map(int, input().rsplit()))
    papers.append(row)


def check(row, col, n):
    global minus_cnt, zero_cnt, plus_cnt
    curr = papers[row][col]

    for i in range(row, row + n):
        for j in range(col, col + n):
            if papers[i][j] != curr:
                next_n = n // 3
                check(row, col, next_n)  # 1
                check(row, col + next_n, next_n)  # 2
                check(row, col + (2 * next_n), next_n)  # 3
                check(row + next_n, col, next_n)  # 4
                check(row + next_n, col + next_n, next_n)  # 5
                check(row + next_n, col + (2 * next_n), next_n)  # 6
                check(row + (2 * next_n), col, next_n)  # 7
                check(row + (2 * next_n), col + next_n, next_n)  # 8
                check(row + (2 * next_n), col + (2 * next_n), next_n)
                return

    if curr == -1:
        minus_cnt += 1
    elif curr == 0:
        zero_cnt += 1
    elif curr == 1:
        plus_cnt += 1
    return

check(0, 0, n)

print(minus_cnt)
print(zero_cnt)
print(plus_cnt)
```

9
[[0 0 0 1 1 1 -1 -1 -1],
[0 0 0 1 1 1 -1 -1 -1],
0 0 0 1 1 1 -1 -1 -1
1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0
0 1 -1 0 1 -1 0 1 -1
0 -1 1 0 1 -1 0 1 -1
0 1 -1 1 0 -1 0 1 -1