

Nhóm 1:

```
1  def fib(n):
2      f = [1, 2]
3      for i in range(2, n + 1):
4          f.append(f[i - 1] + f[i - 2])
5      return f[n - 1]
6
7  n = int(input())
8  print(fib(n))
```

Nhóm 3:

Nhóm 4:

```
1  def solve(n):
2      if n <= 2:
3          return n
4      a = [0] * n
5      a[0] = 1
6      a[1] = 2
7      for i in range(2, n):
8          a[i] = a[i - 1] + a[i - 2]
9      return a[-1]
10
11 n = int(input())
12 res = solve(n)
13 print(res)
```

Nhóm 5:

```

1  def clamber(n):
2      prev = 1
3      curr = 2
4      for i in range(2,n):
5          next = prev + curr
6          prev = curr
7          curr = next
8      print(curr)
9
10 if __name__ == '__main__':
11     n = int(input())
12     if n == 1:
13         print(1)
14     elif n == 2:
15         print(2)
16     else:
17         clamber(n)

```

Nhóm 6:

```

1  def stair(n):
2      a = []
3      if (n <= 1):
4          return 1
5
6      if (n ==2):
7          return 2
8      a.insert(0,1)
9      a.insert(1,1)
10     a.insert(2,2)
11     for i in range(3, n+1, 1):
12         temp = a[i-1] + a[i-2]
13         a.insert(i, temp)
14     return a[n]
15
16 # If main
17 n = int(input())
18 print(stair(n))

```

Nhóm 7:

```
1 def fibo(n):
2     f = [1, 2]
3     for i in range(2, n + 1):
4         f.append(f[i - 1] + f[i - 2])
5     return f[n - 1]
6 n = int(input())
7 print(fibo(n))
```

Nhóm 8:

```
def staircase(n):
    steps = [0] * n
    if n == 1:
        return 1
    elif n == 2:
        return 2
    steps[-1] = 1
    steps[-2] = 2
    for i in range(-3, -(n+1), -1):
        steps[i] = steps[i+1] + steps[i+2]
    return steps[0]

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    n = int(input())
    print(staircase(n))
```

Nhóm 9:

```

1  def findNumberOfWaysToClimbTheTop(N):
2      if N == 1:
3          return 1
4
5      if N == 2:
6          return 2
7
8      numberOfWays = [0] * N
9
10     numberOfWays[0] = 1
11     numberOfWays[1] = 2
12
13     for i in range(2, N):
14         numberOfWays[i] = numberOfWays[i - 1] + numberOfWays[i - 2]
15
16     return numberOfWays[N - 1]
17
18
19 if __name__ == "__main__":
20     numberOfStairs = int(input())
21     print(findNumberOfWaysToClimbTheTop(numberOfStairs))

```

Nhóm 10:

```

1  n = int(input())
2
3  def stair(n):
4      lst = [1, 1, 2]
5      for i in range(3, n + 1):
6          lst.append(lst[i - 1] + lst[i - 2])
7      return lst[n]
8
9  print(stair(n))

```

Nhóm 11:

Nhóm 12:

Nhóm 13:

```

1  def climbStairs(n):
2      if n == 0:
3          return 0
4      elif n == 1:
5          return 1
6      elif n == 2:
7          return 2
8      else:
9          dp = [0] * (n + 1)
10
11         dp[0] = 0
12         dp[1] = 1
13         dp[2] = 2
14
15         for i in range(3, n + 1):
16             dp[i] = dp[i - 1] + dp[i - 2]
17
18         return dp[n]
19 n = int(input())
20 print(climbStairs(n))

```

Nhóm 14:

```

1  n = int(input())
2
3  def stair(n):
4      lst = [1, 1, 2]
5      for i in range(3, n + 1):
6          lst.append(lst[i - 1] + lst[i - 2])
7      return lst[n]
8
9  print(stair(n))

```

Nhóm 15:

```
1  n = int(input())
2
3  def count(n):
4      if n==0 or n ==1:
5          return n
6      return count(n-1) + count(n-2)
7
8  def countSteps(s):
9      return count(s + 1)
10
11 print(countSteps(n))
```

**Nhận xét:** Đa số các nhóm đều làm bài tập tốt và hoàn thành đầy đủ. Nhóm 3,11,12 chưa làm bài tập

Nhóm 15 nộp từ thứ tư tuy nhiên chưa được full điểm do các bạn dùng mỗi đệ quy nên bị quá thời gian so với test case của tụi mình.

Bài này phải giải theo bottom-up thì mới có thể đủ thời gian để thỏa hết các test case.

Ý tưởng thì giống như các nhóm đã làm.

Công thức hồi quy:  $F(n)=F(n-1)+F(n-2)$

Khởi tạo các giá trị cơ sở.

Tạo mảng để lưu các giá trị đã được tính toán từ công thức trên. Sau đó trả về giá trị theo yêu cầu