```
def
       pascal - triangle (n)
                                              Dascal - triangle (4)
        if n==1
                                      result is being memoized.
             return CC133
                                    Meroised, Saving Mcurrive
        Cnd
                                  V Calls.
        result = puscal - triangle (n-1)
        new-row = [1]
        Previous_row = result [-1]
                                      result = [[13, [1,13, [1,2,13]]
       (O... Previous - row. length -1). Cach - with - index do lide!
             Sum = 0
            Sum = result[-1][idx] + result[-1][idx +1]
            new-row. Push (sum)
      end
      new-row. push (1)
      result. Push (hew-row)
                            result = [[13, [1,1], [1,2,1], [1,3,3,13]
ene
```

```
Dascal - triangle (3)
       pascal - triangle (n)
        if n == 1
             return CC133
        Cnd
        result = puscal - triangle (n-1)
        new-row = [1]
                                       result = CC17, C1, 13]
       Previous_row = resul+[-1]
       (O... Previous - row. length -1). Cach - with - index do lide1
             Sum = 0
            Sum = result[-1][idx] + result[-1][idx +1]
            new-row. Push (sum)
      end
      new-row. push (1)
      result. push (hew-row)
end
                           result = [[13,51,13,51,21]]
```

```
def
      pascal - triangle (n)
                                           Dascal - triangle (2)
       if n==1
            return CC133
        Cnd
        result = pascal-triangle(n-1)
       new-row = [1]
                                   result = LIJ]
       Previous_row = result [-1]
      (O ... Previous - row. length -1). Cach - with - index do lide!
            Sum = 0
            Sum = result[-1][idx] + result[-1][idx +1]
           new-row. Push (sum)
      end
      new-row. push(1)
      result. Push (hew-row)
                           result = [ [13, [1, 1]]
end
```

pascal - triangle (n) Cnd result = Puscal - triangle (n-1) new-row = [1] previous_row = result [-1] (O... Previous - row. length -1). Cach - with - index do lide1 Sum = 0 Sum = result[-1][idx] + result[-1][idx +1] new-row. Push (sum) end new-row. push(1) regult. Push (hew-row)

ene