Second Max

```
class Solution:
    # @param A : list of integers
    # @return an integer
    def solve(self, A):
        max_ele=-1
        #found the largest element
        for item in A:
            if item>max_ele:
                max_ele=item

        #find the second largest element
        sec_max=-1
        for item in A:
            if item>sec_max and item!=max_ele:
                sec_max=item

        return sec_max
```

Remove that

```
def main():
    n=int(input()) #length
    A=list(map(int,input().split())) #list
    x=int(input()) #position for which you need to delete
    index=x-1 #actual index which will be deleted
    A.pop(index)#deletes elements at that index

    for i in A:
        print(i,end=" ")

    return 0

if __name__ == '__main__':
    main()
```

```
Find output:

for i in range(-6, -10, -1):

    print(i, end =" ")
```

```
-6 -7 -8 -9
```

Reverse

```
#Your Code Goes Here

# "2 3 4"

input=input().split() #["2","3","4"]

n=int(input[0]) #"2" -> 2

li=list(map(int,input[1:])) #[3,4]


for i in range(n-1,-1,-1):

    print(li[i],end=" ")
```

Difference of odd and even

```
def even_odd(A):

    diff = 0

    # Write your code here

    even_sum=0

    odd_sum=0

    for element in A:

        if element%2==0:

            even_sum+=element

        else:
            odd_sum+=element
        diff=even_sum-odd_sum
    return diff
```

Shopping List

```
def shopping_list():
  shoplist=[] #empty
  while 1: #infinite loop
    item=input() #get an input in string end
    if item == "end": #true
        break
    else:
        shoplist.append(item) #["egg","banana","ice cream"]
  print(shoplist) #["egg","banana","ice cream"]

  '''Calling the function'''
shopping_list()
```

Break :)

9:57 pm - 10:15 pm

2D list
Print row by row

```
def main():
    N, M = map(int, input().split()) #"2 3" -> n=2, m=3
    mat = []
    for i in range(N): #iterating on no of rows
        row = list(map(int, input().split())) #1d List in input
        mat.append(row) #appending the 1d list to the 2D list

    for r in range(N):
        for c in range(M):
            print(mat[r][c],end=" ")
        print() #new line after each row


    return 0

if __name__ == '__main__':
    main()
```

Linear Search - Multiple Occurences

```
class Solution:
    # @param A : list of integers
    # @param B : integer
    # @return an integer
    def solve(self, A, B):
        count=0

        for i in range (len(A)):
            if A[i] == B:
                count+=1

        return count
```

Add matrices

```
class Solution:
    # @param A : list of list of integers
    # @param B : list of list of integers
    # @return a list of list of integers
    def solve(self, A, B):
        C=[]
        for r in range(len(A)):
            temp=[]
            for c in range(len(A[r])):
                temp.append(A[r][c]+B[r][c])
            C.append(temp)

        return C
```

Check if Array is Sorted

```
class Solution:
    # @param A : list of integers
    # @return an integer
    # 1 2 3 4 5 5 3
    def solve(self, A):
        sorted=1

        for i in range(1,len(A)):
            if A[i]<A[i-1]:
                sorted=0
                break

        return sorted
```

Wave pattern printing

```
def main():
    # DO NOT CHANGE THE CODE BELOW.
    global ii
    n = inp[ii: ii + 1][0]
    ii += 1
    mat = [[0 for i in range(n)] for j in range(n)]
    for i in range(n):
        for j in range(n):
            mat[i][j] = inp[ii: ii + 1][0]
            ii += 1
    # DO NOT CHANGE THE CODE ABOVE

    # YOUR CODE GOES BELOW HERE.
    # USE THE GIVEN MATRIX mat FOR ALL THE OPERATIONS.
    for row in range(n):
        for col in range(n):
            if row%2==0: #even -> l to r
                print(mat[row][col],end=" ")

            else: #odd -> r to l
                print(mat[row][n-1-col],end= " ")




    return 0
    # row: 0 1 2 3
    # col: 0 1 2 3
    #    0 1 2
    #0-> 1 2 3 row=0 col=0 1 2
    #1-> 4 5 6 row=1 col=2 1 0
    #2-> 7 8 9

    #n=3, col=0 -> n-1-col=3-1-0=2
    #n=3, col=1 -> n-1-col=3-1-1=1
    #n=3, col=2 -> n-1-col=3-1-2=0

if __name__ == '__main__':
    main()
```