```
'''
Starting at 9:05 pm
Enjoy the song!

Agenda
    •   Map Method
    •   Introduction to 2D list
    •   Indexing in 2D list
    •   Iterating in 2D list
    •   Input in a 2D list
    •   Finding an element, max element and min element
    •   List Comprehension



#Right Shift an Array
'''
Write a program that reads an integer array A from input and
modifies the array by shifting each element to the right by one
position and by shifting the last element to the first position.
After performing the shifting operation, print the modified array.
Note: Ensure there is a space character (' ') at the end of the
line.
'''


#Step 1: take input in string format
n=int(input())
items=input().split() # "2 3 4 5 6" -> ["2","3","4","5","6"]

#Step 2: COnvert each string into integer
A=[]
for i in range(len(items)):
  A.append(int(items[i])) #[2,3,4,5,6]

print(A)
#Step 3: Right shift by 1

#[2,3,4,5,6] -> [2,2,2,2,2] left to right wrong approach

#[2,3,4,5,6] -> [6,2,3,4,5] right to left

last_item=A[-1] #last_item=6
for i in range(len(A)-1,0,-1): #[2,2,3,4,5]
  A[i]=A[i-1]

A[0]=last_item #[6,2,3,4,5]

print(A)
```

```
5
2 3 4 5 6
[2, 3, 4, 5, 6]
```

```
    [6, 2, 3, 4, 5]
```

```
# Slicing A[start:end:jump]
B=[2,3,4,5,6]
B=[B[-1]]+B[:-1] #-1 last item + before the last item
print(B)
```

```
[6, 2, 3, 4, 5]
```

```
#Map Method
# input -> "2 3 4 5 6"
# input().split() -> "2","3" ,"4", "5", "6"
# map(int,input().split()) -> map(int,"2","3" ,"4", "5", "6" ) -> 2,3, 4, 5, 6
# list(map(int,input().split())) -> [2,3, 4, 5, 6]

numbers=list(map(int,input().split()))
print(numbers)
```

```
2 3 4 5 6
[2, 3, 4, 5, 6]
```

```
#Introduction to 2D list
maths=[1,2,1] #1D list
science=[2,3,2] #1D list
history=[3,4,3] #1D list

subjects=[maths,science,history] #2D
print(subjects)
#  c1 c2 c3
# [1, 2, 1] row 1
# [2, 3, 2] row 2
# [3, 4, 3] row 3
```

```
[[1, 2, 1], [2, 3, 2], [3, 4, 3]]
```

```
#Quiz 1
#What type of data structure is subjects?
list of lists
```

```
#Indexing in a 2D list
#  c0 c1 c2
# [1, 2, 1] row 0
# [2, 3, 2] row 1
# [3, 4, 3] row 2

# First index -> choose the row
# Second index -> choose the column

print(subjects[0][2]) #1
print(subjects[1][1]) #3
print(subjects[2]) #[3, 4, 3]
```

```
1
3
[3, 4, 3]
```

```
#Quiz 2
#What will subjects[2][0] give for the same list?
print(subjects[2][0]) #3
```

```
3
```

```
#Iterating in a 2D list
#len(subjects) -> number of rows
#len(subjects[i]) -> number of cols

#subjects: [[1, 2, 1], [2, 3, 2], [3, 4, 3]]
#               0          1          2
# len(subjects) -> 3
# len(subjects[0]) -> [1, 2, 1] -> 3
# len(subjects[1]) -> [2, 3, 2] -> 3
# len(subjects[2]) -> [3, 4, 3] -> 3

#Row-wise
for r in range(len(subjects)): # r -> 0,1,2
  for c in range(len(subjects[r])): # c -> 0,1,2
    print(subjects[r][c], end=" ") #(2,0),(2,1),(2,2)
  print()
```

```
1 2 1
2 3 2
3 4 3
```

```
#Col-wise
for c in range(len(subjects[0])):
  for r in range(len(subjects)):
    print(subjects[r][c], end=" ")
  print()
```

```
1 2 3
2 3 4
1 2 3
```

```
#Input in a 2D list
def take_list_as_an_input():
  li=list(map(int,input().split())) #input for 1 D list
  return li

n=int(input("enter the number of rows"))
two_D_list=[] #2D list
for i in range(n):
  two_D_list.append(take_list_as_an_input()) #appends 1D list at the end

print(two_D_list)
```

```
#[[4 5 6 3 2 ],[5 1],[4 5 7 2 1],[5 8 9 0 1 2 2 4 5 6 7] ]
```

```
enter the number of rows4
4 5 6 3 2
5 1
4 5 7 2 1
5 8 9 0 1 2 2 4 5 6 7
[[4, 5, 6, 3, 2], [5, 1], [4, 5, 7, 2, 1], [5, 8, 9, 0, 1, 2, 2, 4, 5, 6, 7]]
```

```
#Quiz 3
#list(map(int, "4 5 6".split())) will give:

[4,5,6]
```

```
#Problem - Row Wise Sum
# 4 7 10
print(subjects)

#sum(list_name) -> helps you get the sum of a 1D list

for r in range(len(subjects)): # r->0
  row_sum=sum(subjects[r]) #sum([1, 2, 1]) -> 4
  print(row_sum)
```

```
[[1, 2, 1], [2, 3, 2], [3, 4, 3]]
4
7
10
```

Break: 10:06 pm - 10:16 pm

```
#Find Element in a list
li=list(map(int,input().split()))
value=int(input("Which value do you want to search for?"))

if value in li: #helps you check if value is available in li
  print("Found at", li.index(value)) #index gives you the index of value
else:
  print("Not found")
```

```
1 2 3 4
Which value do you want to search for?0
Not found
```

```
#Find max in a list
#Method 1
li = [-13, -53, -23, -21, -55]
print(max(li)) #max returns the maximum value in the list
print(min(li)) #min returns the minimum value in the list

#Method 2
```

```
#item=li[i]
max_value=li[0]
for item in li:
  if max_value<item:
    max_value=item

print(max_value)
```

```
-13
-55
-13
```

```
#iterables -> range, list, set, dict
# range -> numbers
# list -> elements are their in a list
#iterator for loop


#List Comprehension

squares=[] #empty list

for i in range(1,6):
  squares.append(i**2)

print(squares)
```

```
[1, 4, 9, 16, 25]
```

```
sq=[ i**2 for i in range(1,6)]
print(sq)
```

```
[1, 4, 9, 16, 25]
```

```
e=[i for i in range(1,11) if i%2==0]
print(e)

#[ what to place range of values condition]
```

```
[2, 4, 6, 8, 10]
```

```
pairs=[ (i,j) for i in range(3) for j in range(i)]
print(pairs)


for i in range(3): # i -> 0, 1, 2
  for j in range(i): # j -> x, 0, 0, 1
    pairs.append(i,j)
```

```
[(1, 0), (2, 0), (2, 1)]
```

```
#Quiz 4
#What will [i for i in range(5) if i % 2 != 0] give?
p=[i for i in range(5) if i % 2 != 0]
print(p)
```

```
[1, 3]
```

```
#explansion of the above quiz
a=[]
for i in range(5): # 0 1 2 3 4
  if i % 2 != 0: # 1 3
    a.append(i)

print(a)
```

```
[1, 3]
```

```
#Quiz 5
n = 4
li = [i * 2 for i in range(1, n) if i % 2 != 0]
print(li)
```

```
[2, 6]
```

```
#expansion
li=[]
for i in range(1, n):
  if i % 2 != 0:
    li.append(i*2)

print(li)
```

```
#Quiz 6
li = [(i, j) for i in range(3) for j in range(3) if i != j]
print(li)
```

```
[(0, 1), (0, 2), (1, 0), (1, 2), (2, 0), (2, 1)]
```

```
li=[]
for i in range(3): # i -> 0, 1, 2
  for j in range(3): # j -> 0, 1, 2
    if i != j:
      li.append((i, j))
#(0,1),(0,2),(1,0), (1,2), (2,0), (2,1)
```

Double-click (or enter) to edit

```
a = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
```

```
b = [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9]]
#c = [[2, 4, 6], [8,10,12], [14,16,18]] #result

#c[i][j]=a[i][j]+b[i][j]

c=[]
for i in range(len(a)): # i -> 0, 1, 2
  temp=[] #[]
  for j in range(len(a[i])): # j -> 0, 1, 2
    temp.append(a[i][j]+b[i][j]) #[14,16,18]
  c.append(temp) #[[2, 4, 6],[8,10,12],[14,16,18]]

print(c)
```

```
[[2, 4, 6], [8, 10, 12], [14, 16, 18]]
```