List Inbuilt Functions

```
# Max function
# Min function
# Sum function
# Count
# Index
# Reverse
# L + L1
# In operator in list
```

```
runs = [10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
total = 0
for i in runs:
    total += i
print(total)
```

```
907
```

```
# Functions
```

```
def list_sum(li):
    total = 0
    for i in runs:
        total += i

    # return total
    return total
```

```
print(list_sum(runs))
```

```
907
```

```
list_sum(runs)
```

```
907
```

```
# Consecutive elements
```

```
l = [1, 2, 3, 5, 1, 2, 4, 1, 1, 2, 3]
```

```
for i in range(len(l) - 1):
    if l[i] == l[i + 1]:
        print(i)
```

```
7
```

```
def consecutive(li):
    n = len(li)
    for i in range(n - 1):
```

```
        if li[i] == li[i + 1]:
            return i

print(consecutive(l))
```

```
7
```

```
# Runs scored in even index matches

runs = [10, 99, 100, 50, 80, 200, 150, 20, 99, 99]

for i in range(len(runs)):
    if i % 2 == 0:
        print(runs[i])
```

```
10
100
80
150
99
```

```
# Jump statement in range

for i in range(0, len(runs), 2):
    print(runs[i])
```

```
10
100
80
150
99
```

Taking list as input

- map

```
n = int(input())
li = list(map(int, input().split()))
```

```
 3
 1 2 3 4 5 6
```

```
li
```

```
[1, 2, 3, 4, 5, 6]
```

```
li = li[:n]
```

```
li
```

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

List slicing

```
# List slicing doesnt affect the current list
```

```
runs
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs[:]
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
# It is just accessing a data at index = 24
# runs[24]
```

```
runs
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs[0:24]
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
# Our list slicing is handling errors as well
```

```
runs
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
# In case of contradiction it will give empty lists
runs[2:2]
[]
runs[4:2]
[]
runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
runs[::]
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
# Following are even index matches
runs[::2]
[10, 100, 80, 150, 99]

# Odd index matches
runs[1::2]
[99, 50, 200, 20, 99]
runs[1:len(runs):2]
[99, 50, 200, 20, 99]

# Slicing works only on indexes

runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
runs[-1]
99
len(runs)
10
# negative index corresponding to 0th index
```

```
runs[-len(runs)]
```

```
10
```

```
runs[0]
```

```
10
```

```
runs
```

```
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs = [90, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
# start is 0th index or 1st element
# end is len(runs)
```

```
runs[-10:]
```

```
[90, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs
```

```
[90, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs[0:-1]
```

```
[90, 99, 100, 50, 80, 200, 150, 20, 99]
```

```
# -1 -> len(runs) - 1
```

```
runs[1:-1]
```

```
[99, 100, 50, 80, 200, 150, 20, 99]
```

```
# +ve and -ve indexing are just references to the positions
```

```
runs[-1]
```

```
99
```

```
runs[len(runs) - 1]
```

```
99
```

```
# -ve jump
```

```
runs
```

```
[90, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
runs[-len(runs)]
```

```
90
```

```
runs[-1:0:-1]
```

```
[99, 99, 20, 150, 200, 80, 50, 100, 99]
```

```
runs[-1: -len(runs) : -1]
```

```
[99, 99, 20, 150, 200, 80, 50, 100, 99]
```

```
runs[-1 : -len(runs) - 1: -1]
```

```
[99, 99, 20, 150, 200, 80, 50, 100, 99, 90]
```

```
# -10 - 1
```

```
runs
```

```
[90, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
# Getting the reverse of list
```

```
runs[::-1]
```

```
[99, 99, 20, 150, 200, 80, 50, 100, 99, 90]
```

```
# Quiz
```

```
l = [10, 2, 5, 3, 6]
l[::-2]
```

```
[6, 5, 10]
```

```
l = [10, 2, 5, 3, 6]
l[0:5:2]
```

```
[10, 5, 6]
```

```
l = [2, 4, 5, 7, 8]
print(l[5:0])
```

```
[]
```

```
nums = [1, 1, 2, 3, 5, 8, 13]
```

```
nums[1:5]
```

```
[1, 2, 3, 5]
```

```
nums = [0, 25, 50, 75, 100]
# The result of evaluating  nums[0:5:2] is [25, 75]
nums[0:5:2]
```

```
[0, 50, 100]
```

```
# Sum of odd index elements
# Sum of even index elements
# Sum of Sachin scores in first 5 matches, last 5 matches

# Odd index elements
```

```
runs = [10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```

```
sum(runs)
```

```
907
```

```
# even index elements
```

```
runs[::2]
```

```
[10, 100, 80, 150, 99]
```

```
sum(runs[::2])
```

```
439
```

```
# sum of odd index elements
```

```
runs[1::2]
```

```
[99, 50, 200, 20, 99]
```

```
sum(runs[1::2])
```

```
468
```

```
439 + 468
```

```
907
```

```
# Sum of runs in first 5 matches
runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
runs[:5]
[10, 99, 100, 50, 80]
total = sum(runs[:5])
total
339
```

```
# Sum of last 5 matches
runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]

runs[-5:]
[200, 150, 20, 99, 99]
sum(runs[-5:])
568
```

Rotate the array

```
n = [1, 2, 3, 4, 5]
# output = [5, 1, 2, 3, 4]
n[-1] # single value i,e integer
5
```

```
n[:4] # list
```

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

```
new = [n[-1]] + n[:4]
```

```
new
```

```
[5, 1, 2, 3, 4]
```

```
# Final code
```

```
li = list(map(int, input().split()))

li = [li[-1]] + li[:4]
print(li)
```

```
 1 2 3 4 5
```

```
[5, 1, 2, 3, 4]
```

```
li = list(map(int, input().split()))

li = [li[-1]] + li[:len(li) - 1]
print(li)
```

```
 1 2 3 4 5
```

```
[5, 1, 2, 3, 4]
```

```
l = [1, 2, 3]
l1 = [5, 6, 7]
```

```
l + l1
```

```
[1, 2, 3, 5, 6, 7]
```

```
# [1, 2, 3] + 21
```

```
# Doubts
```

```
runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
runs[0:-1:1]
[10, 99, 100, 50, 80, 200, 150, 20, 99]
runs[0:2:-1]
[]
runs[0:-1:1]
[10, 99, 100, 50, 80, 200, 150, 20, 99]
runs[-1]
99
runs[-2]
99
runs[-3]
20

runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
runs
[10, 99, 100, 50, 80, 200, 150, 20, 99, 99]
```