⇒ For some problem, you have to take input of 45 values

You have to also use all of them later.

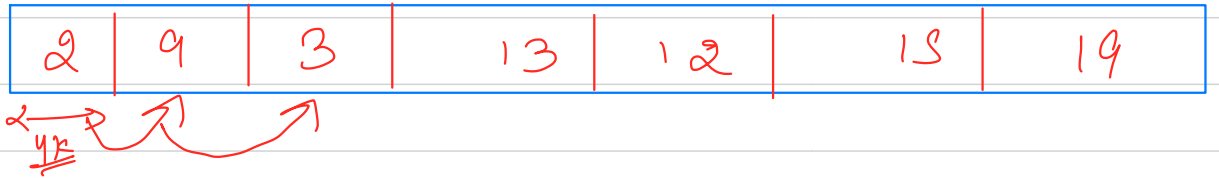
↳ To solve this issue we have arrays.

You have to store names of 200 people

Formal definition → Arrays are data structure which can store data in linear (contiguous) fashion, and occupies a linear space in memory.

continuous

representable of array



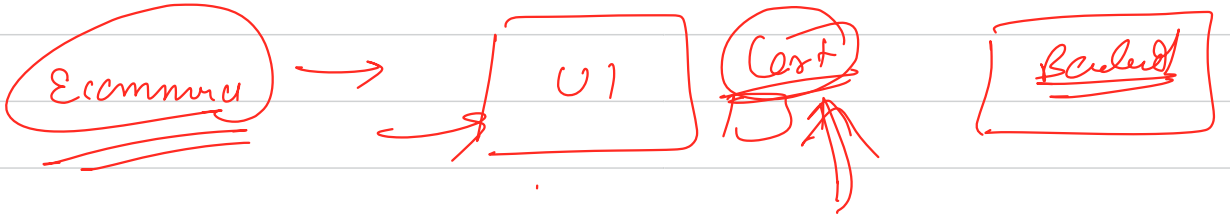
How arrays can be useful ??

Real life application

arrays → have a big size

200 names → array of size 200

You won't have to make 200 variables



In flowcharts we create arrays by writing

<name of array> = [] ⟶ empty array

a = [] ⟶ empty

a = ["mayo", "apple", "grape"] ⟶ 3 length array

Initialize an array

```
temp = [1]
```

`temp` → name / label of the variable

`[1]` → this is an array with a value 1 inside it

→ It will create an array of length 1

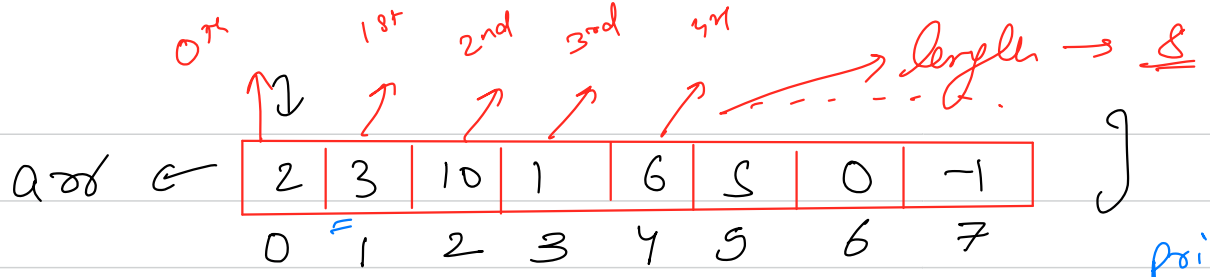
```
numbers = [2, 5, 1, 7, 10, 11]
```

How to access elements

```
x=10
print(x)
```

Concept of indexing / positioning.

What is indexing ?? The process of giving unique positions to all the elements of an array is called indexing in array



print(arr[1]) → 3

print(arr[6]) → 0

print(arr[8])

↳ error

indexing 0 – 7

Generally in programming indexing starts with 0.

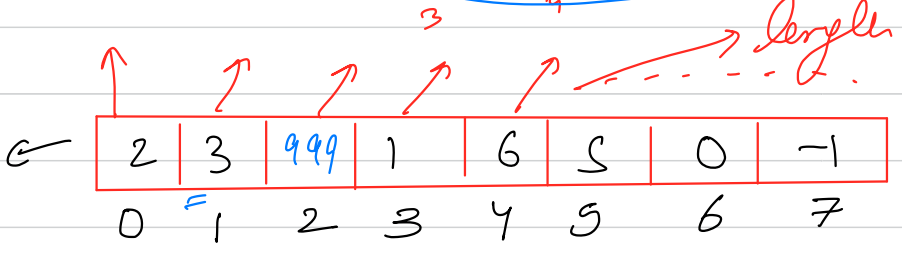
Ex C, C++, Java, Python, Ruby, Javascript, C#, etc

Best ones are exceptions → Ex Mat lab (indexing starts with 1)

print (arr[1])

update → arr[2] = 999

access



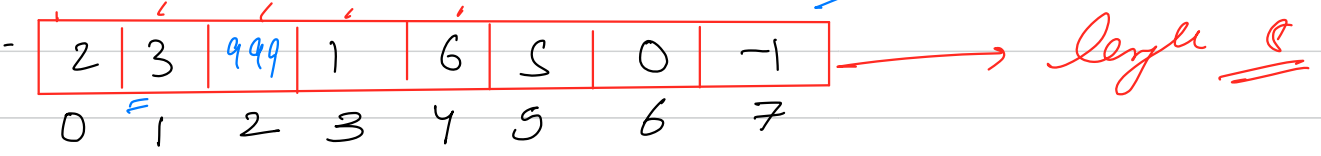
Initialize → arr = []

arr = [1, 2]

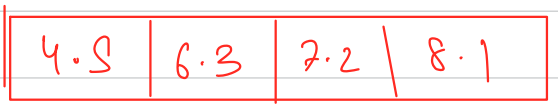
print → print (arr[0])
print (arr[1])

update any index → arr[2] = 999

we want to initialize this array in memory



Syntax → arr = [2, 3, 999, 1, 6, 5, 0, -1]



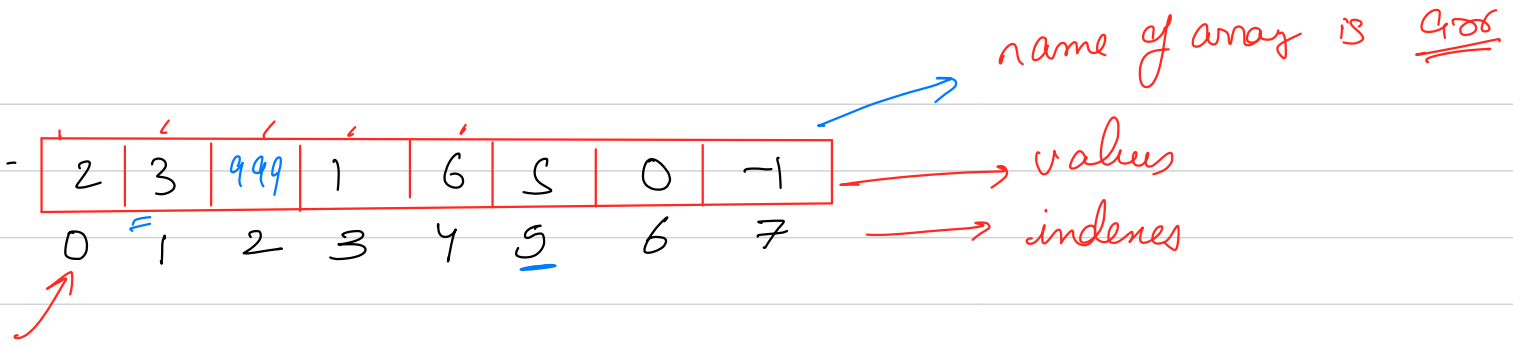
arr

length 4

arr = [4.5, 6.3, 7.2, 8.1]

arr = [ ] → empty

↘ length 0



How to print value at the 5th index ?

print (arr[5])

→ indexing → print (arr[index])

valid value of index

[0, 7]

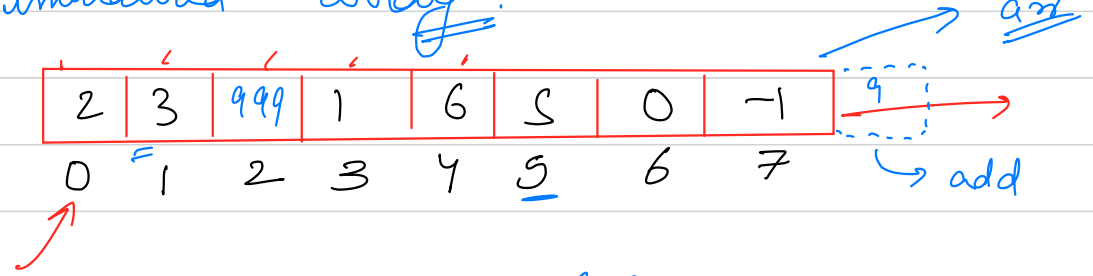
general → [0, length(arr) - 1]

Ex arr[-1] → error

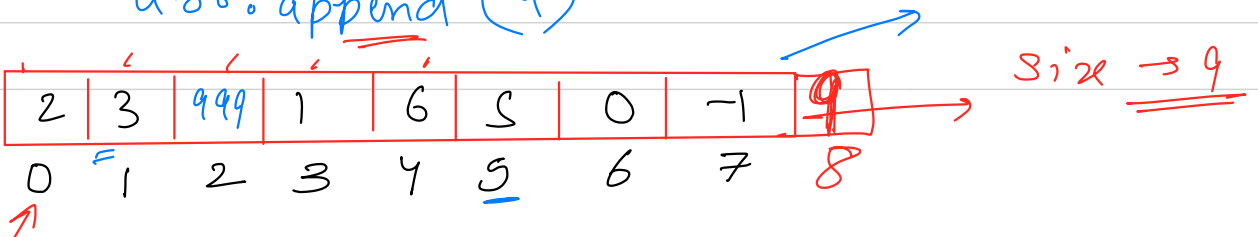
How to update value at 3rd index?

arr[3] = 19

How to further add some values in already initialized array?



arr.append(9)



arr.append(6)

arr.append(5)

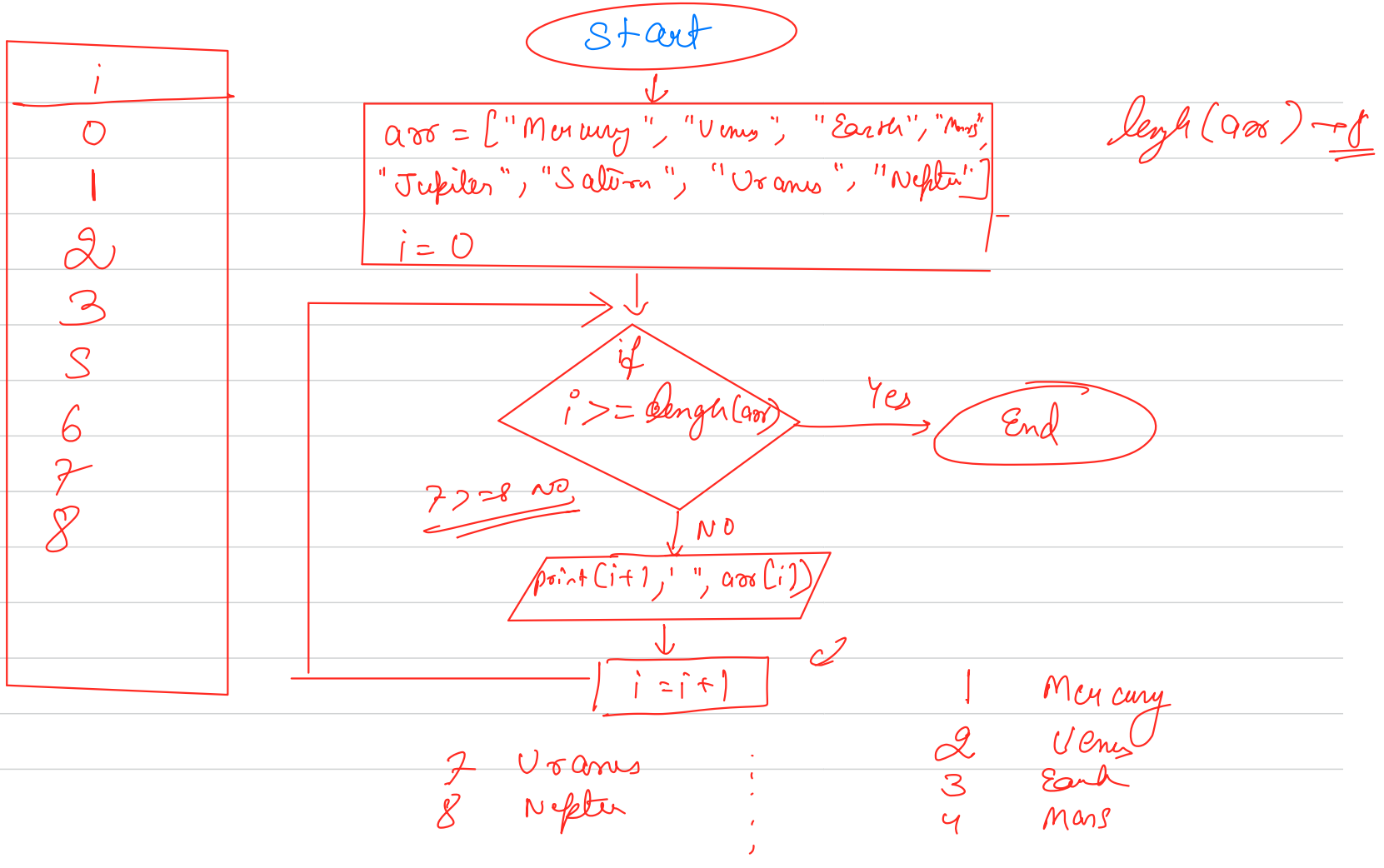
n values → loop

```
x = input()
arr.append(x)
```

→ adds to last

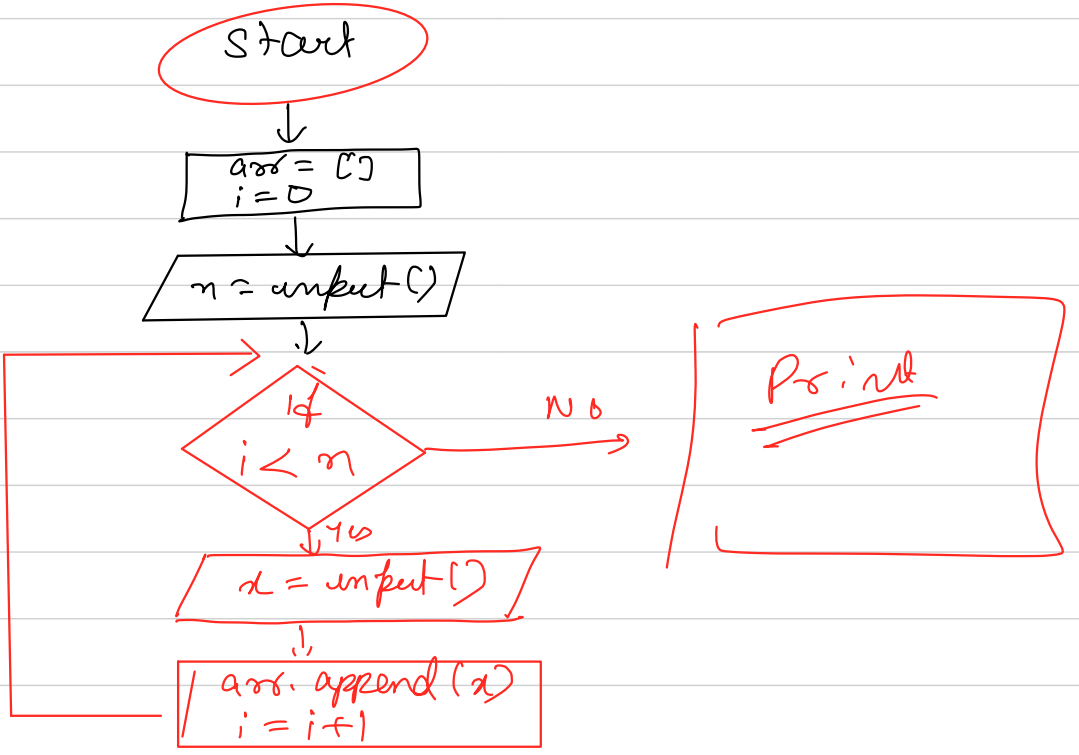
Q⇒ Draw a flowchart to print

1 Mercury
2 Venus
3 Earth
4 Mars
5 Jupiter
6 Saturn
7 Uranus
8 Neptune

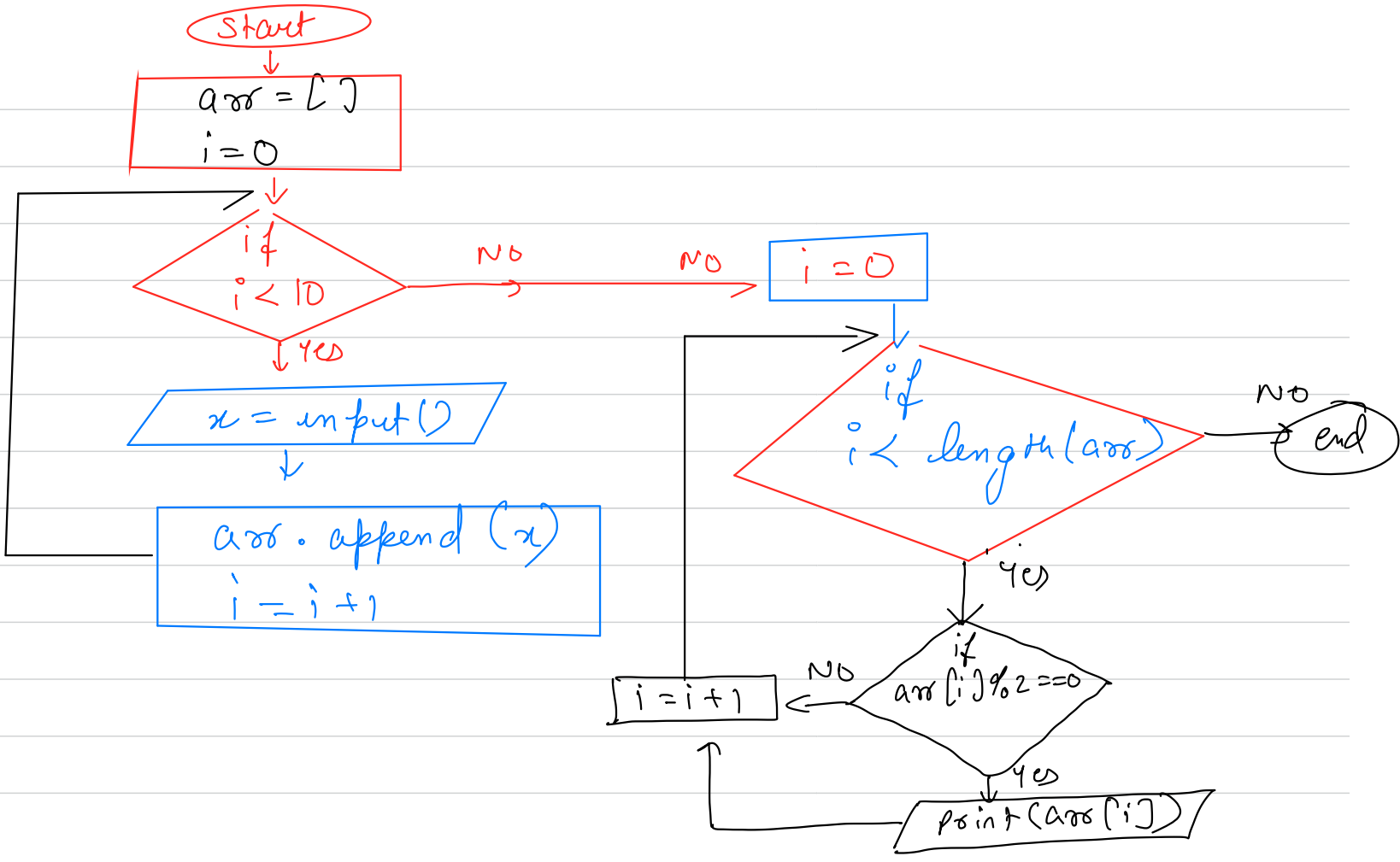

Start
arr = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"]
i = 0
length(arr) → 8
if
i >= length(arr)
Yes
End
7 >= 8 NO
NO
print(i+1, " ", arr[i])
i = i+1
i
0
1
2
3
4
5
6
7
8
1 Mercury
2 Venus
3 Earth
4 Mars
7 Uranus
8 Neptune

Q) User will give you input of n student names. Print all of them but first store in an array

append ( )

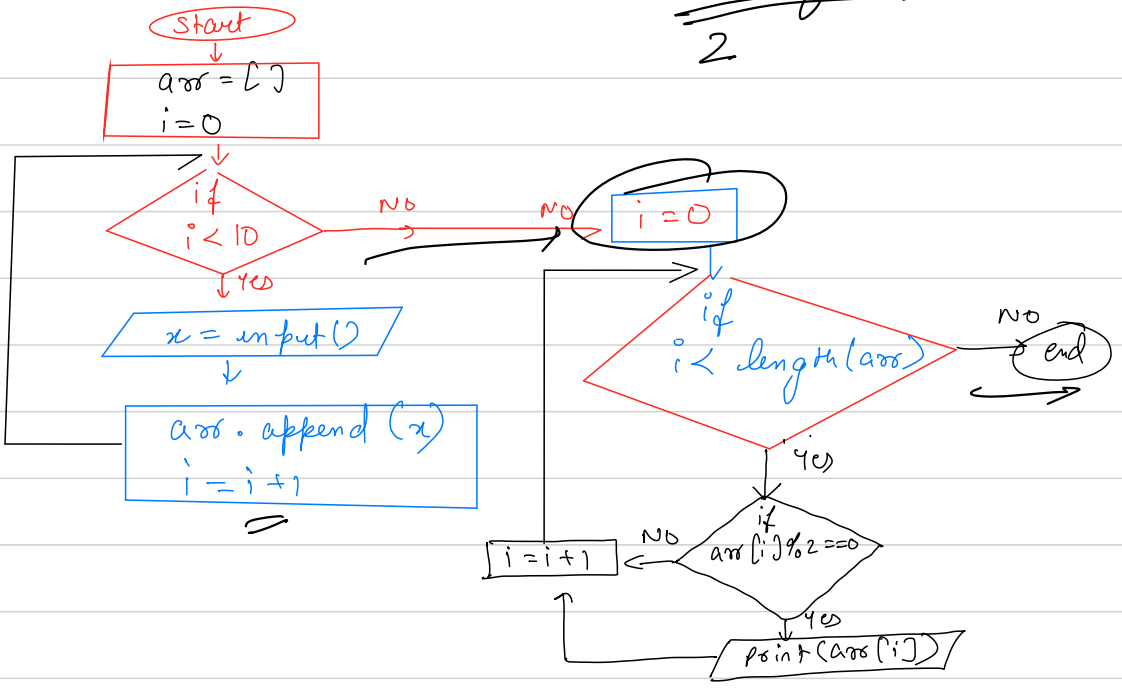


Q-> Draw a flowchart to store 10 numbers from user in an array and then print those numbers which are even.


Start
arr = [ ]
i = 0
if
i < 10
NO
Yes
x = input()
arr . append (x)
i = i + 1
NO
i = 0
if
i < length(arr)
NO
end
Yes
if
arr [i] % 2 == 0
NO
i = i + 1
Yes
print (arr [i])

display

2



i	arr	x
0		
1		
2		
3	]	
.		
.		
.		
.		
.		
.		

UI ⟹ [2, 5, 7, 10, 12, 13, 14, 15, 19, 29]

0 1 2 3 4 5 6 7 8 9