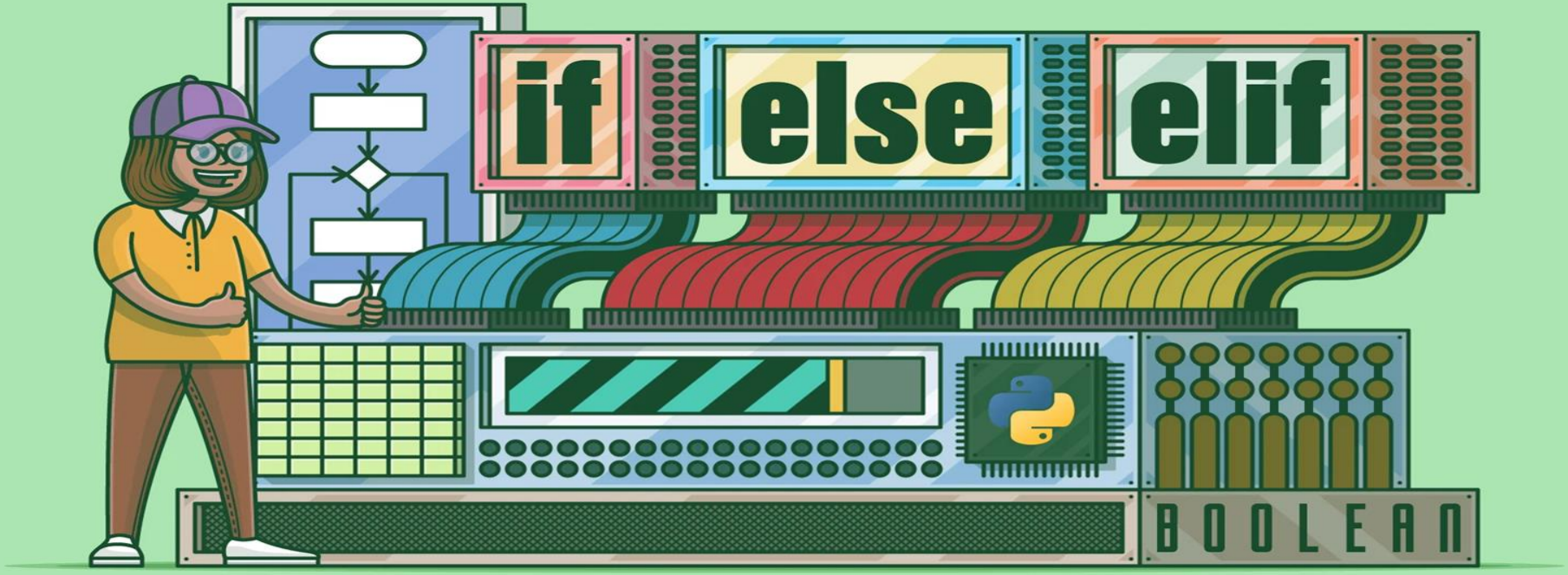
PYTHON

Lecture - 05

Python Control Statements

- conditional statements
- loops

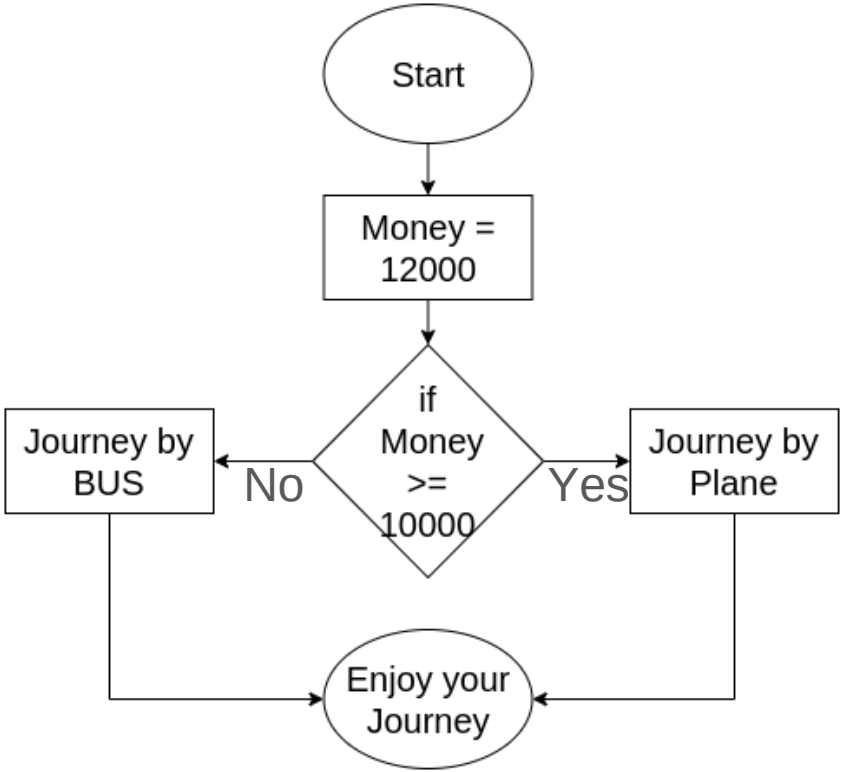
Decision Making Statements



What is Decision Making?

Decision making means what should we do in a circumstances.

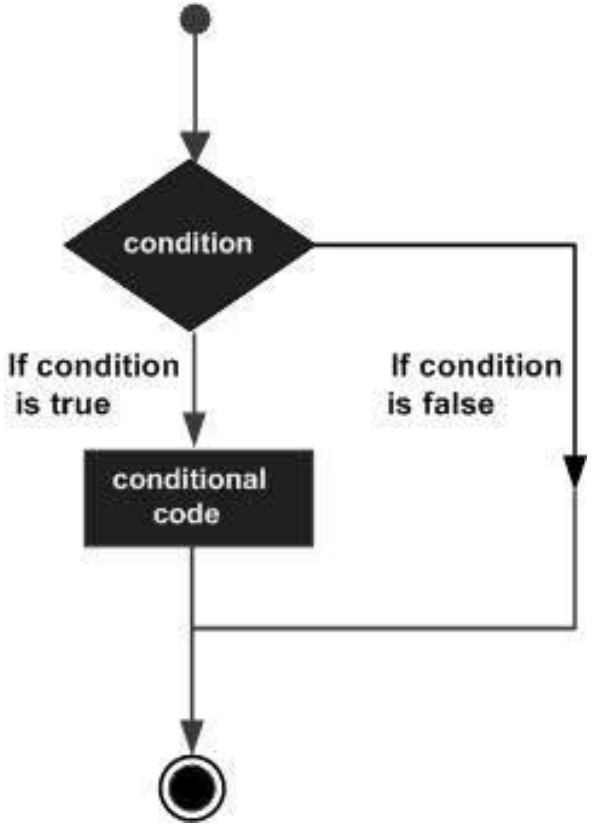
The flowchart helps you to decide what should you choose in the time of travel.

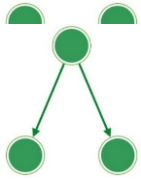


Python Decision Making (if Statement)

if Statement Syntax

```
if expression:
    statement(s)
```





Python Decision Making (if Statement)

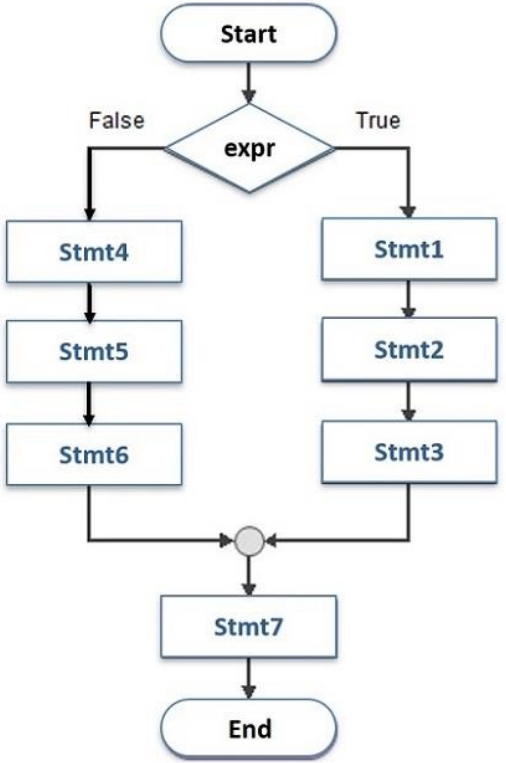
Let's consider an example of a customer entitled to 10% discount if his purchase amount is > 1000. if not, then no discount is applicable. Given his purchase amount, find out the payable amount

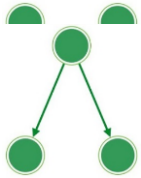
```
purchaseAmount = 1250
payableAmount = purchaseAmount
if(purchaseAmount>1000):
   payableAmount -= purchaseAmount*0.1

print(payableAmount)
```

Python Decision Making (if-else Statement)

if-else Statement Syntax

```
if expression1:
   statement(s)
elif expression2:
   statement(s)
else:
   statement(s)
```

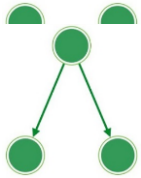




Python Decision Making (if-else Statement)

You can vote if your age is more or equal to 18. Now, given an age, find if s/he can vote or not?

```
age = 20
if age >= 18:
   print("s/he can vote")
else:
   print("s/he can't vote")
```



Python Decision Making (if-else Statement)

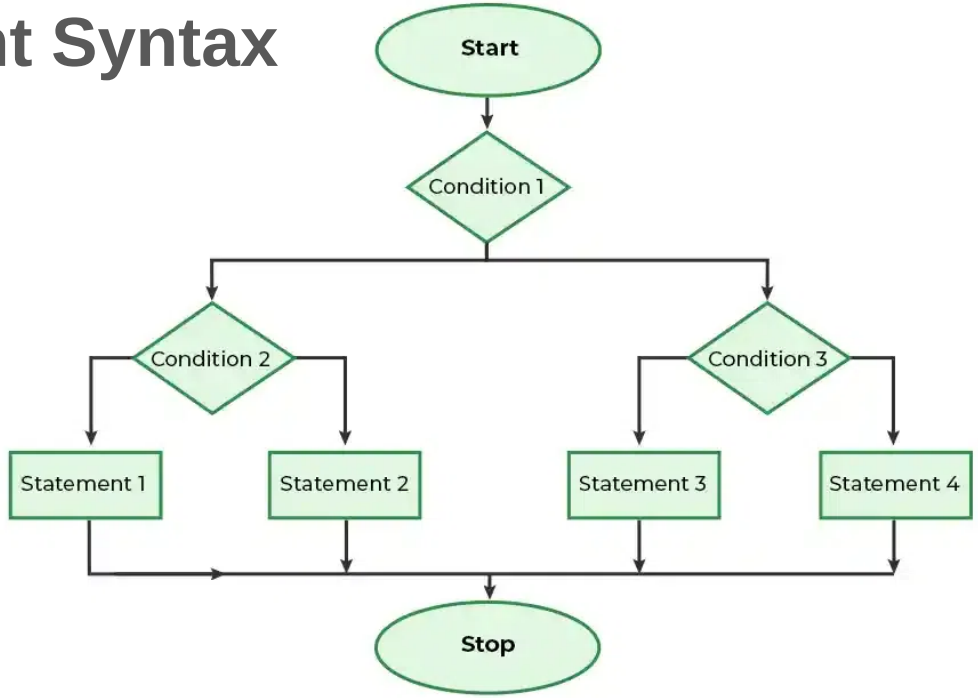
You are given 3 numbers. Find out the largest number.

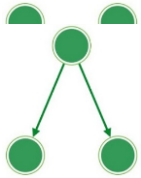
```
a,b,c = map(int,input().split(" "))

if(a>=b and a>=c):
   print(a)
elif(b>=c):
   print(b)
else:
   print(c)
```

Python Decision Making (Nested if Statement)

Nested if Statement Syntax

```
if expression1:
   statement(s)
   if expression2:
      statement(s)
   else:
      statement(s)
elif expression3:
   statement(s)
else:
   statement(s)
```

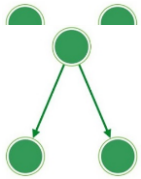




Python Decision Making (Nested if Statement)

You are given 3 numbers. Find out the largest number.

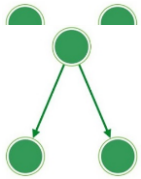
```
a,b,c = 5,7,2
if a>b:
   if a>c:
       print(a)
   else:
       print(c)
elif b>c:
   print(b)
else:
   print(c)
```



Python Decision Making (Match-Case Statement)

Nested if Statement Syntax

```
match variable_name:
    case 'pattern 1' : statement 1
    case 'pattern 2' : statement 2
    ...
    case 'pattern n' : statement n
```



Python Decision Making (Match-Case Statement)

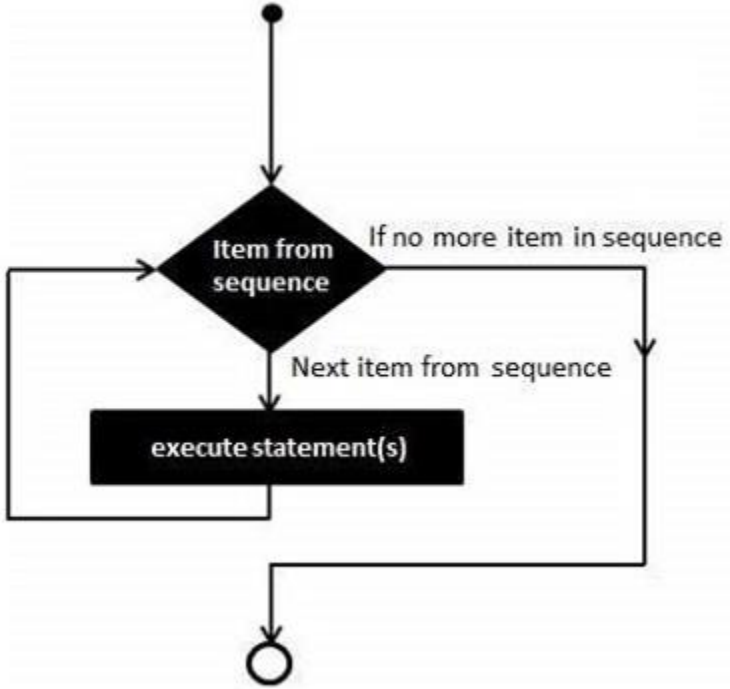
The following code has a function named weekday(). It receives an integer argument, matches it with all possible weekday number values, and returns the corresponding name of day.

```
def weekday(n):
  match n:
      case 0: return "Monday"
      case 1: return "Tuesday"
      case 2: return "Wednesday"
      case 3: return "Thursday"
      case 4: return "Friday"
      case 5: return "Saturday"
      case 6: return "Sunday"
      case _: return "Invalid day number"
print (weekday(3))
print (weekday(6))
print (weekday(7))
```


DEFINITE
FOR
1 to 10
BREAK
CONTINUE
1... 2... 3... 4... 5... 6... 10
1
2
3
4
5
6
7
8
9
10
END

Python For Loop (Syntax)

```
for iterating_var in sequence:
    statements(s)
```



Python For Loop (Example - 1)

Iterate over a List and print it’s elements.

```
cities = ['Barisal', 'Dhaka', 'Khulna',
'Sylhet']
for city in cities:
   print(city)
```

OUTPUT

Barisal
Dhaka
Khulna
Sylhet

Python For Loop (Example - 2)

Given last 5 days temperature in a list. Find its average.

```
temp = [35.8, 39.6, 38.2, 40.8, 41.2]
sum = 0.0
for x in temp:
    sum+=x

avg = sum/len(temp)
print(avg)
```

OUTPUT

39.12

Python For Loop (Example - 3)

Print all chars of a given string

```
s = "university"
for i in s:
   print(i)
```

OUTPUT

u
n
i
v
e
r
s
i
t
y

Python For Loop with range() Function

`range(start, stop, step)` → returns a List of int datatype

Parameter	Description
start	Optional. An integer number specifying at which position to start. Default is 0
stop	Required. An integer number specifying at which position to stop **(not included)**.
step	Optional. An integer number specifying the incrementation. Default is 1

range(2,10,2) → 2, 4, 6,8
range(6) → 1, 2, 3, 4, 5

Python For Loop (Example-01)

Iterate over a List and print it’s elements.

```
cities = ['Barisal', 'Dhaka', 'Khulna',
'Sylhet']
for i in range(len(cities)):
  print(cities[i])
```

OUTPUT

Barisal
Dhaka
Khulna
Sylhet

Python For Loop (Example - 2)

Given last 5 days temperature in a list. Find its average.

```
temp = [35.8, 39.6, 38.2, 40.8, 41.2]
sum = 0.0
for i in range(len(temp)):
  sum+=temp[i]
 avg = sum/len(temp)
print(avg)
```

OUTPUT

39.12

Python For Loop (Example - 3)

Print all chars of a given string

```
s = "university"
for i in range(len(s)):
  print(s[i])
```

OUTPUT

u
n
i
v
e
r
s
i
t
y

Python Loop Control Statement

Sr.No.	Control Statement & Description
1	**break statement** Terminates the loop statement and transfers execution to the statement immediately following the loop.
2	**continue statement** Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
3	**pass statement** The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

Python Loop Control Statement (break)

Find if the given list has an even value or not.

```
List = [3, 5, 6, 7, 10]
ok = 0
for x in List:
   if x%2==0:
       ok = 1
       break

if ok:
   print("Found")
else:
   print("Not Found")
```

OUTPUT

Found

Python Loop Control Statement (continue)

Given a list. Print all odd numbers of the list.

```
List = [3, 5, 6, 7, 10]
ok = 0
for x in List:
  if x%2==0:
     continue
  print(x)
```

OUTPUT

```
3
5
7
```

Python Loop Control Statement (pass)

It is a null operation; nothing happens when it executes. Python pass statement is also useful in places where your code will eventually go, but has not been written yet.

```
s = "university"
for i in range(len(s)):
  pass
```

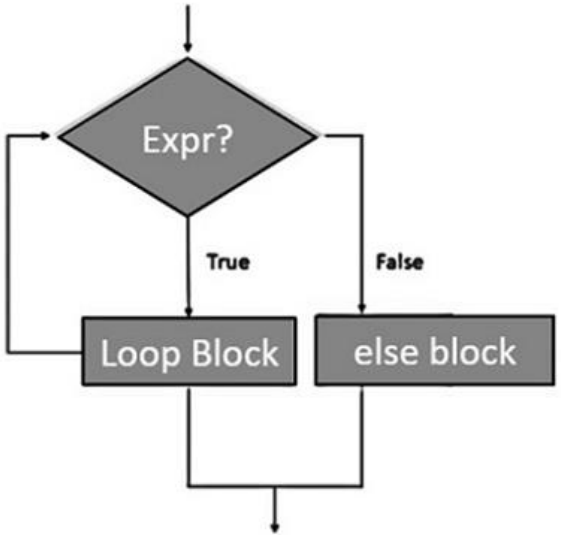
OUTPUT

#The code print nothing.

`for` loops cannot be empty, but if you for some reason have a `for` loop with no content, put in the `pass` statement to avoid getting an error.

Python for-each Loop (Syntax)

```
for variable_name in iterable:
 #stmts in the loop
 .
 .
else:
 #stmts in else clause
 .
 .
```



Python for-each Loop (Example)

Find if the given list has an even value or not.

```
List = [3, 5, 11, 7, 9]
ok = 0
for x in List:
  if x%2==0:
     print("Found")
     break
else:
  print("Not Found")
```

OUTPUT

Not Found

Note: The `else` block will NOT be executed if the loop is stopped by a `break` statement.

Python for-each Loop (Example)

Find if the given list has an even value or not.

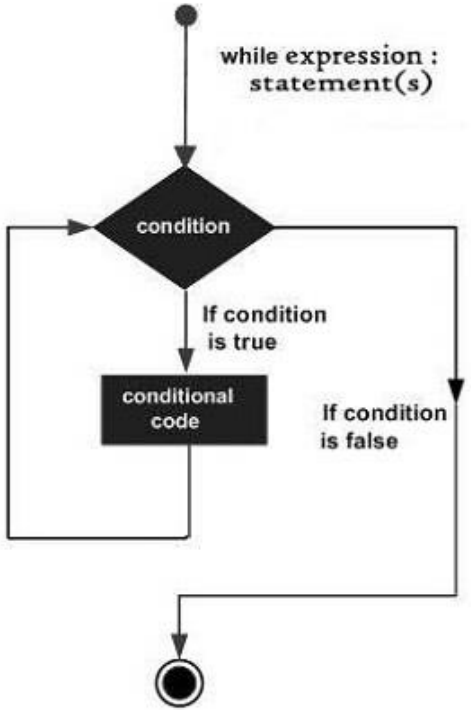
```
List = [3, 5, 10, 7, 9]
ok = 0
for x in List:
  if x%2==0:
     print("Found")
     break
else:
  print("Not Found")
```

OUTPUT

Found

Python While Loop (Syntax)

```
initialization
while expression:
    statement(s)
        inc/dec
```



Python While Loop (Example - 1)

Iterate over a List and print it's elements.

```
cities = ['Barisal', 'Dhaka', 'Khulna',
'Sylhet']
i = 0
while(i<len(cities)):
  print(cities[i])
  i+=1
```

OUTPUT

Barisal
Dhaka
Khulna
Sylhet

Python While Loop (Example - 2)

Given last 5 days temperature in a list. Find its average.

```
temp = [35.8, 39.6, 38.2, 40.8, 41.2]
sum = 0.0
i = 0
while i<len(temp):
  sum+=temp[i]
  i+=1
avg = sum/len(temp)
print(avg)
```

OUTPUT

39.12

Python While Loop (Example - 3)

Print all chars of a given string

```
s = "university"
i = 0
while i<len(s):
  print(s[i])
  i+=1
```

OUTPUT

u
n
i
v
e
r
s
i
t
y

Python Nested Loop (Syntax)

```
for iterating_var in sequence:
   for iterating_var in sequence:
      statements(s)
   statements(s)
```

Python Nested Loop (Example - 1)

Print all prime numbers upto 100

```
num = 2
while(num <= 100):
  j = 2
  while(j <= (num/j)):
     if not(num%j):
        break
     j = j + 1
  if(j > num/j):
     print(num,"is prime")
  num = num + 1
```

OUTPUT

```
2 is prime
3 is prime
5 is prime
…
…
89 is prime
97 is prime
```

Exercise Time

Exercise – Bonus_01

- Find the sum of numbers from **1** to **5**.
- Find the average of numbers from **1** to **10**.
- Given **N**. Find the sum of numbers from **1** to **N**.
- Print all numbers for 10 to 1.
- Take input as n and check whether it is prime or not.
- Print all prime numbers between 100 to 200.

Exercise – 4.1

Firstly take **N** and than **N** strings as input. Strings can be both uppercase and lowercase. Concat them and print a single string in Camelcase.

Input:
3
MaX
paYAblE
AmouNT

Output:
maxPayableAmount

Hints: use upper() and lower() function.

“abc”.upper() → “ABC”
“ABC”.lower() → “abc”

Exercise – 4.1 (ans)

```
n = int(input())
List = []
for i in range(n):
  s = input()
  List.append(s)

ans = ""
for s in List:
  ans+=s[:1].upper()
  ans+=s[1:].lower()

ans = ans[:1].lower() + ans[1:]
print(ans)
```

Exercise – 4.2

Write a Python program that computes the greatest common divisor (GCD) of two positive integers.

Input:
8
12

Output:
4

Exercise – 4.2 (ans)

```
x = 86
y = 44
n = min(x,y)
gcd = 0
for i in range(1,n+1):
    if x%i == 0 and y%i == 0:
        gcd = i

print(f"The GCD of {x} and {y} is {gcd}.")
  # It's called formatted string, which helped to print
  # variables with strings.
```

Exercise – 4.3

Write a Python program to find the least common multiple (LCM) of two positive integers.

Input:
8
12

Output:
24

Exercise – 4.3 (ans)

```
num1 = 8
num2 = 12
for i in range(max(num1, num2), 1 + (num1 * num2), max(num1, num2)):
   if i % num1 == i % num2 == 0:
       lcm = i
       break

print("LCM of", num1, "and", num2, "is", lcm)
```

Exercise – 4.4

Take a List as input and find its MEX.
MEX is smallest non-negative number that isn't present in the list.

Input_1: [2, 1, 4, 0, 5]
Output_1: 3

Input_2: [2, 1, 4, 0, 5, 3]
Output_2: 6

Hints: you can sort the List using the function sort(), then find out the mex using a loop, but how? Think yourself.

A = [5, 2, 4]
A.sort()
print(A) → [2, 4, 5]

Exercise – 4.4 (ans)

```
arr = [2, 1, 4, 0, 5, 1]
N = len(arr)

arr.sort()
mex = 0
for idx in range(N):
    if arr[idx] == mex:
        mex += 1

print(mex)
```

Exercise – 4.5

You are given a phone number as a string. Report as **“INV”** if the number is invalid. Otherwise find out its operator company name. Every valid number has exact 11 digits and first 3 digits decide it’s operator company name. Here is the list of valid operators & their codes:
Grameenphone → ‘017’ or ‘013’
Teletalk → ‘015’
Banglalink →’014’ or ‘019’
Airtel →’016’
Robi →’018’

Sample Input: "01411111111"
Sample Output: “Banglalink”

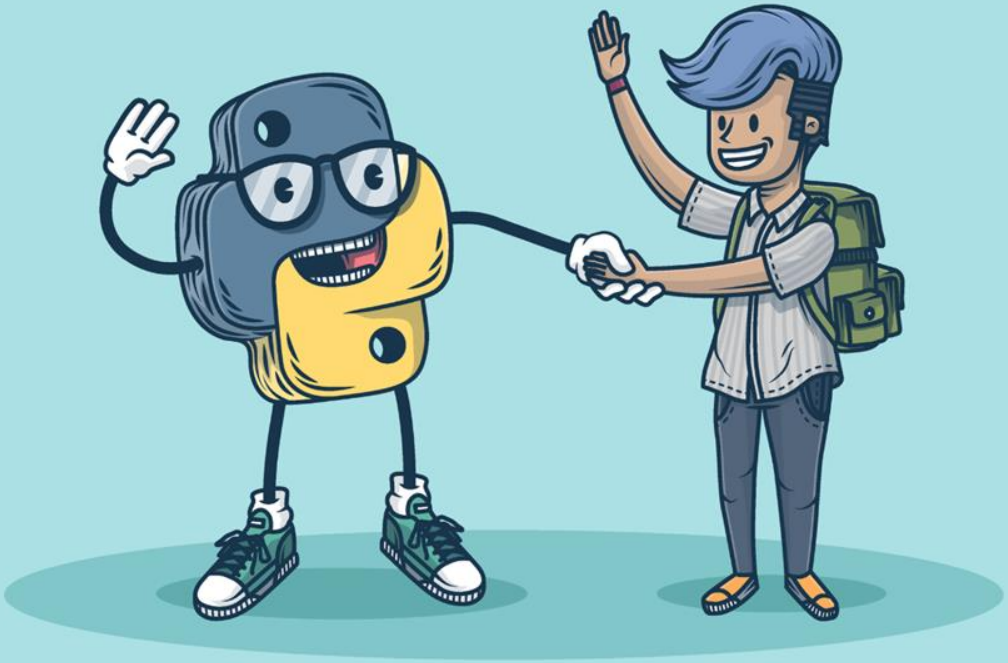
Solve the Problem using match case

Exercise – 4.5 (ans)

```
num = input()
if(len(num)!=11):
  print("INV")

match num[:3]:
    case "017":
        print("Grameenphone")
    case "013":
        print("Grameenphone")
    case "014":
        print("Banglalink")
    case "019":
        print("Banglalink")
    case "015":
        print("Teletalk")
    case "016":
        print("Airtel")
    case "018":
        print("Robi")
    case _:
        print("INV")
```

Resources

- https://www.tutorialspoint.com/python/index.htm
- https://www.w3resource.com/python/python-tutorial.php
- https://www.w3resource.com/python-exercises/string/
- https://www.w3schools.com/python/
- https://www.geeksforgeeks.org/python-programming-language/
- https://youtu.be/t2_Q2BRzeEE?si=OO6J_YNCZykedqsT
- https://realpython.com/
- Head First Python, 3rd Edition by Paul Barry
- Automate the Boring Stuff with Python By Al Sweigart.



Thank You