অধ্যায়-৭

লুপিং
(Looping)

৭.০ ভূমিকা (Introduction) ঃ

যে-কোনো প্রোগ্রামিং ল্যাংগুয়েজ-এর একটি গুরুত্বপূর্ন বিষয় হলো লুপিং (Looping)। লুপ মানে হলো বার বার। যখন একই কাজ বার বার করার দরকার পড়ে তখন লুপ ব্যবহার করতে হয়। কন্ডিশনাল লজিকের একটি অন্যতম উদাহরণ হলো এই লুপ। কারণ লুপে কন্ডিশনের উপর ভিত্তি করে একই কাজ বার বার করা হয়। উক্ত অধ্যায়ে আমরা পাইথনে ব্যবহৃত বিভিন্ন লুপিং (Looping) সম্পর্কিত স্টেটমেন্ট ও তাদের ব্যবহার সম্পর্কিত বিষয়াদি নিয়ে বিশদ আলোচনা করব।

৭.১ কন্ডিশনাল ও আনকন্ডিশনাল লুপিং ফ্লো (Conditional & unconditional looping flow) ঃ

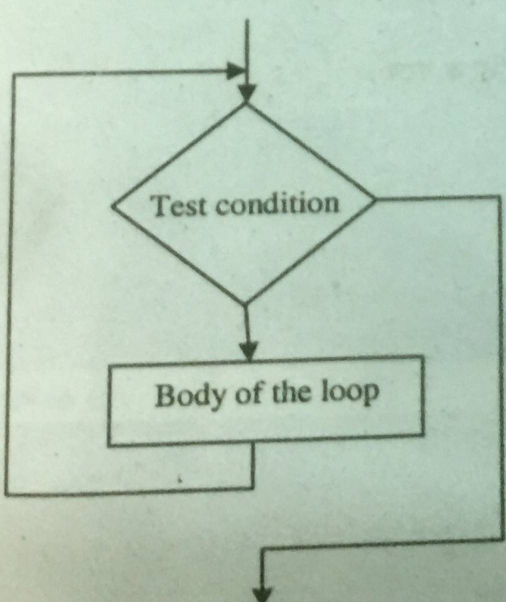
প্রোগ্রামে যদি কখনো নির্দিষ্ট কাজ দুই বা ততোধিক বার সম্পাদনের প্রয়োজন হয় সেক্ষেত্রে স্টেটমেন্ট বারবার না লিখে লুপ কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয়। প্রোগ্রামের কোনো অংশে বিশেষ নির্দিষ্ট সংখ্যক বার বা প্রদত্ত কোনো শর্তে না পৌঁছা পর্যন্ত অথবা অনির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করাকে লুপিং বলা হয়। লুপ দুটি প্রধান অংশ নিয়ে গঠিত, যথা–

- লুপের মূল অংশ (Body of the loop) এবং
- কন্ট্রোল স্টেটমেন্ট (Control statement)

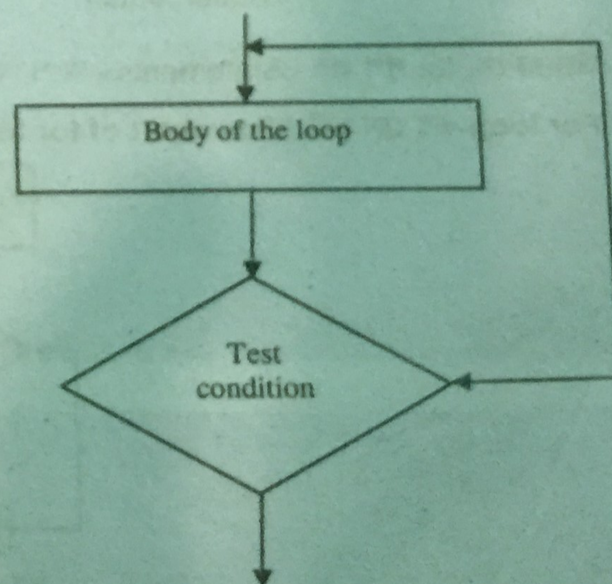
কন্ট্রোল স্টেটমেন্ট লুপের মূল অংশের শুরুতে বা শেষেও থাকতে পারে।

লুপ স্ট্রাকচার দুই ধরণের, যথা–

- এন্ট্রি কন্ট্রোল (Entry control) ও
- এক্সিট (Exit control) কন্ট্রোল



(a) Entry control



(b) Exit control

এন্ট্রি কন্ট্রোল (Entry control) লুপে লুপ বডির নির্বাহ শুরু হওয়ার আগেই টেস্ট কন্ডিশন পরীক্ষা করা হয়। কন্ডিশন সত্য না হলে লুপ বডি সম্পাদিত হয় না। কিন্তু এক্সিট (Exit control) কন্ট্রোল লুপে শর্তহীনভাবে একবার লুপ বডি সম্পাদিত হয়। অতঃপর টেস্ট কন্ডিশন পরীক্ষা করা হয় এবং কন্ডিশন সত্য না হওয়া ভিত্তিতে লুপ বডির পরবর্তী এক্সিকিউশন নির্ভর করে।

লুপিং দুই প্রকার, যথা–

- কন্ডিশনাল লুপিং ফ্লো (Conditional looping flow) ও
- আন কন্ডিশনাল লুপিং ফ্লো (Unconditional looping flow)

কন্ডিশনাল লুপিং ফ্লো (Conditional looping flow) ঃ যখন কোনো প্রোগ্রাম কন্ডিশনাল স্টেটমেন্টের প্রভাবে এক্সিকিউট হয়, তখন তাকে কন্ডিশনাল লুপিং ফ্লো বলে। এতে প্রোগ্রামে ব্যবহৃত কন্ডিশনের সাপক্ষে একই ধরনের কাজ বার বার হয়। পাইথনে মূলত দুই ধরনের কন্ডিশনাল লুপিং ফ্লো স্টেটমেন্ট আছে, যথা–

- for স্টেটমেন্ট
- while স্টেটমেন্ট

আন-কন্ডিশনাল লুপিং ফ্লো (Unconditional looping flow) ঃ যখন কোনো প্রোগ্রাম আন-কন্ডিশনাল স্টেটমেন্টের প্রভাবে এক্সিকিউট হয়, তখন তাকে আন কন্ডিশনাল লুপিং ফ্লো বলে। এতে প্রোগ্রামের কোনো অংশ অসংখ্য বার আবর্তিত হতে থাকে, একে ইনফিনিট লুপও বলা হয়।

৭.২ for লুপ স্টেটমেন্ট (for loop statement in python) ঃ

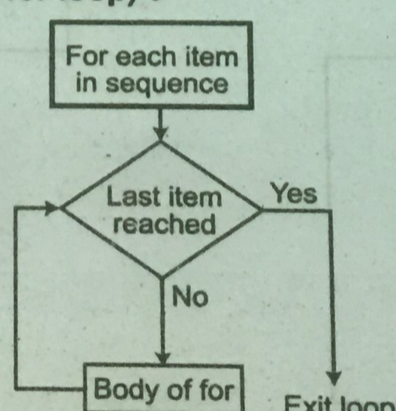
for লুপ স্টেটমেন্ট (for loop statement) ঃ প্রোগ্রামে এক বা একাধিক স্টেটমেন্ট একটি নির্দিষ্ট সংখ্যকবার এক্সিকিউট করার জন্য for লুপ ব্যবহৃত হয়। সাধারণ কোনো ভেরিয়েবল ব্যবহার করে for লুপের আবর্তন সংখ্যা গণনা করা হয়। এরূপ ভেরিয়েবলকে কাউন্টার ভেরিয়েবল বলে। পাইথনে সিকুয়েন্সিয়াল টাইপ অবজেক্ট যেমন– লিস্ট, টাপল, সেট, ডিকশনারি, স্ট্রিং ইত্যাদি ইটারেট করার জন্যই for লুপ ব্যবহৃত হয়।

সিনট্যাক্স (Syntax) ঃ

```
for <Variable> in range(<Number of Times>):
    <Statements>
```

উল্লেখ্য যে, for লুপ এর <Statements> অংশ অবশ্যই ইনডেন্টেশন দিতে হবে।

For loop-এর ফ্লো-চার্ট (Flowchart of for loop) ঃ



চিত্র ঃ Flowchart of for loop

উদাহরণ-১ ঃ

```
for x in range(3):
    print('Hello')
    print('goodbye')
```

উদাহরণ-২ ঃ

```
for x in range(5):
    print(x)
```

উদাহরণ-৩ ঃ $1^2 + 2^2 + 3^2 + \ldots\ldots\ldots\ldots + 9^2 + 10^2 = ?$ নির্ণয়ের প্রোগ্রাম

```
sum = 0
for i in range (1, 11) :
    sum = sum + (i*i)
print ("sum of first 10 squares is", sum)
```

লিস্টের সঙ্গে for লুপের ব্যবহার (for loop with list) ঃ

পাইথনের কোন লিস্টকে for লুপ ব্যবহার করে ইটারেট করা যায়।

উদাহরণ-১ ঃ

```
#  Program to find the sum of all numbers stored in a list.
# List of numbers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
sum = 0
# iterate over the list
for i in numbers:
    sum = sum + i
print ("The sum is", sum)
```

আউটপুট ঃ

```
The sum is 55
```

উদাহরণ-২ ঃ

```
friends = ['Tahsin', 'Mahi', 'Mahdee']
for friend in friends:
    print ('Happy New Year:', friend)
    print ('Thanks a Lot')
```

আউটপুট ঃ

```
Happy New Year: Tahsin
Thanks a Lot
Happy New Year: Mahi
Thanks a Lot
Happy New Year: Mahdee
Thanks a Lot
```

উদাহরণ-৩ ঃ

```
a = {'Name' : 'A.K.M Tanveer ul Islam', 'Nickname' : 'Mahi', 'Address' : 'Comilla'}
print(a)
for item in a:
    print(item)
```

আউটপুট ঃ

```
{'Name': 'A.K.M Tanveer ul Islam', 'Nickname': 'Mahi', 'Address': 'Comilla'}
Name
Nickname
Address
```

স্ট্রিং-এর সঙ্গে for লুপের ব্যবহার (for loop with string) ঃ

পাইথনে কোন স্ট্রিংকে for লুপ ব্যবহার করে ইটারেট করা যায়।

উদাহরণ-১ ঃ

```
Name = 'Mahi'
for letter in Name:
    print(letter)
```

আউটপুট ঃ

```
M
a
h
i
```

উদাহরণ-২ ঃ

```
# Loop over string indexes.
str = "Mahi"
Length = len (str)
for i in range (0, length) :
    print (str [i])
```

আউটপুট ঃ

```
m
a
h
i
```

for লুপের সঙ্গে range() Function-এর ব্যবহার (range() Function with for Loop)ঃ for লুপের সঙ্গে range() Function-কে বিভিন্নভাবে ব্যবহার করা যায়। range() Function এ দুই সেট প্যারামিটার আছে, যথা–

- range(Stop)
- range(Start, Stop, Step)

এখানে

- Start হলো সিকুয়েন্সের ইনিশিয়াল ভ্যালু (শুরুর মান)
- Stop হলো সর্বশেষ মানের পূর্বের মান এবং
- Step হলো কাউন্টার ভেরিয়েবলের ইনক্রিমেন্ট বা ডিক্রিমেন্ট (অর্থাৎ মান কত করে বাড়বে বা কমবে) ভ্যালুকে বুঝায়।

উল্লেখ্য যে,

- range() ফাংশনের সকল প্যারামিটার অবশ্যই ইন্টিজার (পূর্ণ সংখ্যা) হতে হবে।
- range() ফাংশনের প্যারামিটারসমূহ পজিটিভ বা নেগেটিভ যে-কোনো সংখ্যা হতে পারে।
- পাইথনে range() ফাংশনের ইনডেক্স সাধারণত ০ থেকে শুরু হয়।

উদাহরণ-১ ঃ

```
for a in range(5):
        print (a)
```

আউটপুট ঃ

```
0
1
2
3
4
```

উদাহরণ-২ ঃ

```
a = range(1, 10)
for i in a:
    print (i )
```

আউটপুট ঃ

```
1
2
3
4
5
6
7
8
9
```

উদাহরণ-৩ ঃ

```
for a in range(3,6):
    print (a)
```

আউটপুট ঃ

```
3
4
5
```

উদাহরণ-৪ ঃ

```
for a in range(2,10,2):
    print (a)
```

আউটপুট ঃ

```
2
4
6
8
```

উদাহরণ-৫ ঃ

```
for a in range(0,-10,-2):
    print (a)
```

আউটপুট ঃ

```
0
-2
-4
-6
-8
```

নিম্নে কতিপয় range() ফাংশন ও তাদের আউটপুট দেখানো হলো ঃ

range() ফাংশন	আউটপুট
range(10)	0,1,2,3,4,5,6,7,8,9
range(1, 10)	1,2,3,4,5,6,7,8,9
range(1, 10, 2)	1,3,5,7,9
range(10, 0, -1)	10,9,8,7,6,5,4,3,2,1
range(10, 0, -2)	10,8,6,4,2
range(2, 11, 2)	2,4,6,8,10
range(-5, 5)	−5,−4,−3,−2,−1,0,1,2,3,4
range(1, 2)	1
range(1, 1)	(empty)
range(1, -1)	(empty)
range(1, -1, -1)	1,0
range(0)	(empty)

for লুপের সঙ্গে else-এর ব্যবহার (for loop with else) ঃ for লুপের সঙ্গে else-কে ব্যবহার করা যায়। যখন for লুপের ইটারেশন শেষ হয় তখনই else ব্লক এক্সিকিউশন শুরু হয়।

উদাহরণ ঃ

```
for n in range(0, 11):
    print (n)
    n = n + 1
else:
    print('The loop is over.')
```

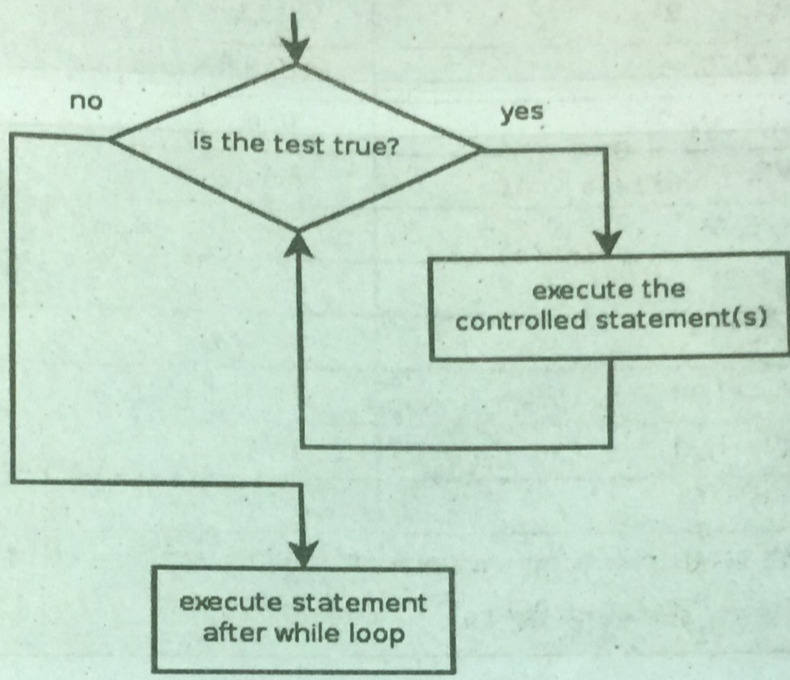
আউটপুট ঃ

```
0
1
2
3
4
5
6
7
8
9
10
The loop is over.
```

৭.৩ while লুপ স্টেটমেন্ট (while loop statement in python) ঃ

একই কোড বার বার রান করানোর জন্য while লুপ ব্যবহার করা হয়। এতে একটি কন্ডিশন ব্যবহার করা হয়, যদি কন্ডিশনটি সত্য হয়, তাহলে while লুপের ভেতরে থাকা কোডগুলো রান হবে। আর যদি কন্ডিশনটি মিথ্যা হয়, তাহলে while লুপের ভেতরের কোডগুলো রান হবে না।

ফ্লোচার্ট (Flowchart) ঃ



চিত্র ঃ while লুপ

সিনট্যাক্স (Syntax) ঃ

while লুপের সাধারণ গঠন হচ্ছে ঃ

```
while Condition:
    Statements
```

while লুপের কন্ডিশনটি যতক্ষণ পর্যন্ত সত্য হবে, ততক্ষণ পর্যন্ত while লুপটি চলবে। কন্ডিশন মিথ্যা হলেই প্রোগ্রামের এক্সিকিউশন শেষ হয়ে যাবে।

উদাহরণ-১ ঃ

```
a = 1
while a < 10:
    print (a)
    a=a+1
print ("This is an end")
```

আউটপুট ঃ

```
1
2
3
4
5
6
7
8
9
This is an end
```

উদাহরণ-২ ঃ

```
n = 1
Sum = 0
while n <= 100:
    Sum = Sum + n
    n = n + 1
print(Sum)
```

আউটপুট ঃ

```
5050
```

উদাহরণ-৩ ঃ

```
a = 0
while a < 10:
    a = a + 1
    print(a)
```

আউটপুট ঃ

```
1
2
3
4
5
6
7
8
9
10
```

উদাহরণ-৪ ঃ

```
a = 1
b = 0
print("Enter numbers to add them : ")
print("press 0 to quit")
while a ! = 0:
    print("Current sum : ",b)
    a = float(input("Enter the numbers: "))
    b = b + a
print("Total sum : ",b)
```

আউটপুট ঃ

```
Enter numbers to add them :
press 0 to quit
Current sum :  0
Enter the numbers: 5
Current sum :  5.0
Enter the numbers: 4
Current sum :  9.0
Enter the numbers: 3
Current sum :  12.0
Enter the numbers: 4
Current sum :  16.0
Enter the numbers: 4
Current sum :  20.0
Enter the numbers: 0
Total sum :  20.0
```

উদাহরণ-৫ ঃ

```
password = str()
while password! = "mahi2015mahdee":
      password = input("Password: ")
print("Welcome, Please Come in")
```

আউটপুট ঃ

```
Password: mahi2015mahdee
Welcome, Please Come in
```

উদাহরণ-৬ ঃ

```
name = input("What is your UserName: ")
password = input("What is your Password: ")
input1 = None
input2 = None
while input1 = name:
      input1 != input("please enter your current username : ")
while input2 ! = password:
      input2 = input("please enter your current password: ")
print("Welcome back to your system")
```

আউটপুট ঃ

```
What is your UserName: mahbub
What is your Password: mahbub123456
please enter your current username : mahbub
please enter your current password: mahbub123456
Welcome back to your system
```

উদাহরণ-৭ ঃ

```
a = 0
b = 1
count = 0
while count = 10:
    count = count + 1
    print(a,b)
    a = a + b
    b = a + b
```

আউটপুট ঃ

```
0 1
1 2
3 5
8 13
21 34
55 89
144 233
377 610
987 1597
2584 4181
```

while লুপের সঙ্গে if এর ব্যবহার (while Loop with if) ঃ while লুপের সঙ্গে if statement কে এক সঙ্গে ব্যবহার করা যায়। যেমন—

```
count = 0
sum = 0.0
number = 1
print("Enter 0 to exit the loop")
while number != 0:
    number = float(input("Enter a number: "))
    if number != 0:
        count = count + 1
        sum = sum + number
    if number == 0:
        print("The average was:", sum / count)
```

আউটপুট ঃ

```
Enter 0 to exit the loop
Enter a number: 10
Enter a number: 5
Enter a number: 7
Enter a number: 14
Enter a number: 0
The average was: 9.0
```

while লুপের সঙ্গে if else ও elif এর ব্যবহার (while loop with if else & elif statement) ঃ while লুপের সঙ্গে else, if else কিংবা elif statement-কে ও ব্যবহার করা যায়। যেমন—

উদাহরণ ঃ

```
number = 7
guess = 0
count = 0
print("Guess the number!")
while guess != number:
    guess = int(input("Enter the number :  "))
    count = count + 1
    if guess == number:
        print("You are right!")
    elif guess < number:
        print("It's bigger")
    elif guess > number:
        print("It's not so big.")
    if count > 3:
        print("You are expired. don't try again")
    else:
        print("Good job!")
```

আউটপুট ঃ

```
Guess the number!
Enter the number :  10
It's not so big.
Good job!
Enter the number :  15
It's not so big.
Good job!
Enter the number :  5
It's bigger
Good job!
```

৭.৪ Infinite, Nested লুপ স্টেটমেন্ট (Infinite and nested loop statement in python) ঃ

Infinite লুপ স্টেটমেন্ট (Infinite loop statement in python) ঃ প্রোগ্রামের এক বা একাধিক স্টেটমেন্টকে শর্তহীনভাবে বা অনির্দিষ্ট সংখ্যক বার আবর্তন করানোর প্রক্রিয়াকে ইনফিনিট (Infinite) লুপিং বলে। পাইথনে দুই ধরনের ইনফিনিট লুপ তৈরি করা যায়, যথা–

- Infinite for লুপ (Infinite for loop)
- Infinite while লুপ (Infinite while লুপ loop)

Infinite for লুপ (Infinite for loop) ঃ for স্টেটমেন্ট ব্যবহার করে যে ইনফিনিট লুপ তৈরি করা হয় তাকে ইনফিনিট for লুপ বলে। যেমন–

```
l = [1]
for x in l:
    l.append(x + 1)
    print(x)
```

পাইথনে উপরোক্ত কোডটি রান করলে ১ থেকে অসীম সংখ্যক সংখ্যা প্রিন্ট হতেই থাকবে। ইনফিনিট for লুপের অ্যাকশন থামতে হলে ctrl+c চাপতে হবে।

Infinite while লুপ (Infinite while loop) ঃ পাইথনে ইনফিনিট লুপ বলতে মূলত Infinite while লুপকেই বুঝায়। while স্টেটমেন্ট ব্যবহার করে যে ইনফিনিট লুপ তৈরি করা হয়, তাকে ইনফিনিট while লুপ বলে। যেমন–

উদাহরণ-১ ঃ

```
while 1 == 1:
    print("Welcome! Don't afraid. Please Press ctrl+c")
```

উদাহরণ-২ ঃ

```
flag = 1
while (flag):
    print ('Given flag is really true!')
print ("Good bye!")
```

ইনফিনিট while লুপের ক্ষেত্রেও ctrl+c চেপে অপারেশন বন্ধ করতে হয় না হলে অনন্তকাল ধরে একই ফলাফল প্রদান করে যেতে থাকবে।

Nested লুপ স্টেটমেন্ট (Nested loop statement in python) ঃ একটি লুপিং স্টেটমেন্টের মধ্যে অন্য কোনো এক বা একাধিক লুপিং স্টেটমেন্ট ব্যবহার করলে তাকে নেস্টেড লুপিং বলে। এক্ষেত্রে মধ্যবর্তী লুপকে ইনার লুপ এবং বহিঃস্থ লুপকে আউটার লুপ বলে। আউটার লুপের পূর্বে ইনার লুপের কাজ শেষ হয়। পাইথনে দুই ধরনের নেস্টেড লুপ স্টেটমেন্ট রয়েছে। যথা–

Nested for লুপ স্টেটমেন্ট ও
Nested while লুপ স্টেটমেন্ট

Nested for লুপ স্টেটমেন্ট (Nested for loop statement) ঃ যখন একটি for লুপ স্টেটমেন্টের মধ্যে অন্য কোন এক বা একাধিক for লুপ স্টেটমেন্ট ব্যবহার করা হয় তখন তাকে নেস্টেড for লুপ বলে। লক্ষণীয় যে, আউটার লুপ কন্ডিশন সত্য হলে ইনার লুপের স্টেটমেন্ট এক্সিকিউশন চলতে থাকে, যতক্ষণ না ইনার লুপের কন্ডিশন মিথ্যা হবে। ইনার লুপের কন্ডিশন মিথ্যা হলে আবার আউটার লুপে যাবে এবং উপরোক্ত নিয়মে প্রক্রিয়াটি চলতে থাকবে যতক্ষণ না উভয় লুপই মিথ্যা হবে।

সিনট্যাক্স ঃ

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

উদাহরণ-১ ঃ

```
num_list = [1, 2, 3]
alpha_list = ['a', 'b', 'c']

for number in num_list:
    print(number)
    for letter in alpha_list:
        print(letter)
```

আউটপুট ঃ

```
1
a
b
c
2
a
b
c
3
a
b
c
```

উদাহরণ-২ ঃ

```
count = 1
for i in range(10):
    for j in range(0, i):
        print(count, end='')
        count = count +1
    print()
input()
```

আউটপুট ঃ

```
1
23
456
78910
1112131415
161718192021
22232425262728
2930313233343536
373839404142434445
```

Nested while লুপ স্টেটমেন্ট (Nested while loop statement) ঃ যখন একটি while লুপের মধ্যে এক বা একাধিক while লুপ ঘোষণা করা হয়, তখন তাকে Nested while লুপ বলে।

সিনট্যাক্স ঃ

```
while Condition:
    while Condition:
        statement(s)
    statement(s)
```

উদাহরণ-১ ঃ

```
b = 0
while b < = 2:
    x = 0
    print (b)
    while x < = 2:
        print (x)
        x + = 1
    b+ = 1
```

আউটপুট ঃ

```
0
0
1
2
1
0
1
2
2
0
1
2
```

উদাহরণ-২ ঃ

```
i = 2
while (i < 20) :
    J = 2
    while (j < = (i/j)):
        if not (i % j):
            break
        j = j + 1
    if (j > i/j):
        print (i, "is prime")
    i = i + 1
```

আউটপুট ঃ

```
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
```

৭.৫ Break, Continue ও Pass স্টেটমেন্ট (Break, continue & pass statement) :

লুপের এক্সিকিউশন কন্ট্রোল করার জন্য লুপ কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয়। পাইথনে তিন ধরনের লুপ কন্ট্রোল স্টেটমেন্ট ব্যবহার করা হয়, যথা–

- Break স্টেটমেন্ট
- Continue স্টেটমেন্ট ও
- Pass স্টেটমেন্ট

Break স্টেটমেন্ট (Break statement) : Break স্টেটমেন্টের কাজ হলো Break স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে লুপ থেকে বের হয়ে যাবে এবং লুপের বাইরে কোনো স্টেটমেন্ট থাকলে সেগুলোর এক্সিকিউশন শুরু করা।

উদাহরণ-১ :

```
for letter in "string":
    if letter  == "n" :
        break
    print(letter )
print ("The end" )
```

আউটপুট :

```
M
a
h
i
The end
```

উদাহরণ-২ :

```
x = 1
while x <= 10:
    if x == 5:
        break
    print(x)
    x += 1
```

আউটপুট :

```
1
2
3
4
```

উদাহরণ-৩ :

```
for i in range (0, 10):
    print (i)
    i = i + 1
    if i == 5:
        break
print ("The end")
```

আউটপুট :

```
0
1
2
3
4
The end
```

Continue স্টেটমেন্ট (Continue statement) ঃ Continue স্টেটমেন্টের কাজ হলো Continue স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে যাবে ঠিকই কিন্তু লুপ থেকে বের হবে না। শুধুমাত্র কন্ডিশনের অংশটুকু বাদ দেবে এবং প্রোগ্রামের বাকি অংশের কাজ সম্পন্ন করবে।

উদাহরণ-১ ঃ

```
for letter in "Mahi & Mahdee":
    if letter  == "&" :
        continue
    print(letter )
print ("The end" )
```

আউটপুট ঃ

```
M
a
h
i


M
a
h
d
e
e
The end
```

উদাহরণ-২ ঃ

```
for number in range(1, 10):
    if number == 6:
        continue
    print(number)
```

আউটপুট ঃ

```
1
2
3
4
5
7
8
9
```

Pass স্টেটমেন্ট (Pass Statement) ঃ Empty লুপ তৈরির জন্য Pass স্টেটমেন্ট ব্যবহৃত হয়। Pass স্টেটমেন্ট ব্যবহার করলে কোনো স্ট্রিং-এর সর্বশেষ ক্যারেক্টার বা লিস্ট এর সর্বশেষ সংখ্যা আউটপুট হিসেবে দেখাবে।

উদাহরণ-১ ঃ

```
for letter in 'Mahi is a good boy':
    pass
print ('Last Letter :', letter)
```

আউটপুট ঃ

```
Last Letter : y
```

উদাহরণ-২ ঃ

```
for Number in range(10):
    pass
print (Number)
```

আউটপুট ঃ

```
9
```

৭.৬ লুপিং স্টেটমেন্টস ব্যবহার করে প্রোগ্রাম (Programs using looping statements) ঃ

প্রোগ্রাম-১ ঃ কোনো একটি পূর্ণসংখ্যা ইনপুট নিয়ে ঐ পূর্ণসংখ্যার নামতা প্রিন্ট করার প্রোগ্রাম।

```
print('Please, input the number:')
number = int(input())
count = 1
while count <= 10:
    print(number, 'x', count, '=', number*count)
    count += 1
```

প্রোগ্রাম-২ ঃ কোনো একটি পূর্ণসংখ্যা ইনপুট নিয়ে এটি মৌলিক সংখ্যা কি না তা নির্ণয়ের প্রোগ্রাম।

```
number = int(input("Enter a number: "))
if number > 1:
    for i in range(2,number):
        if (number % i) == 0:
            print(number,"is not a prime number")
            break
    else:
        print(number,"is a prime number")

else:
    print(number,"is not a prime number")
```

অনুশীলনী-৭

অতি সংক্ষিপ্ত প্রশ্নোত্তর ঃ

১। লুপিং বলতে কী বুঝায়?

উত্তর ঃ প্রোগ্রামের কোনো অংশবিশেষ নির্দিষ্ট সংখ্যক বার বা প্রদত্ত কোনো শর্তে না পৌছা পর্যন্ত অথবা অনির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করাকে লুপিং বলা হয়।

২। লুপিং ব্যবহারের সুবিধা কী?

উত্তর ঃ প্রোগ্রামের নির্দিষ্ট কোনো কাজ দুই বা ততোধিক বার সম্পাদনের প্রয়োজন হলে সেক্ষেত্রে লুপিং স্টেটমেন্ট ব্যবহার করলে একই স্টেটমেন্ট বারবার লেখার প্রয়োজন হয় না।

৩। এন্ট্রি কন্ট্রোল লুপ ও এক্সিট কন্ট্রোল লুপের পার্থক্য কী?

উত্তর ঃ এন্ট্রি কন্ট্রোল (Entry control) লুপে লুপ বডির নির্বাহ শুরু হওয়ার আগেই টেস্ট কন্ডিশন পরীক্ষা করা হয়। কন্ডিশন সত্য না হলে লুপ বডি সম্পাদিত হয় না। অন্যদিকে এক্সিট (Exit control) কন্ট্রোল লুপে শর্তহীনভাবে একবার লুপ বডি সম্পাদিত হয়। অতঃপর টেস্ট কন্ডিশন পরীক্ষা করা হয় এবং কন্ডিশন সত্য না হওয়া ভিত্তিতে লুপ বডির পরবর্তী এক্সিকিউশন নির্ভর করে।

৪। লুপিং এর প্রকারভেদ উল্লেখ কর।

উত্তর ঃ লুপিং দুই প্রকার, যথা–

- কন্ডিশনাল লুপিং ফ্লো (Conditional looping flow) ও
- আনকন্ডিশনাল লুপিং ফ্লো (Unconditional looping flow)

৫। দুটি লুপিং স্টেটমেন্টের নাম লেখ।

অথবা, পাইথনের দুটি লুপ স্ট্যাটমেন্টের নাম লেখ। [বাকাশিবো-২০১৭]

উত্তর ঃ পাইথনে ব্যবহৃত দুটি লুপিং ফ্লো স্টেটমেন্ট হলো ঃ

- for স্টেটমেন্ট এবং
- while স্টেটমেন্ট

৬। for লুপিং স্টেটমেন্ট কখন ব্যবহৃত হয়?

উত্তর ঃ প্রোগ্রামে এক বা একাধিক স্টেটমেন্ট একটি নির্দিষ্ট সংখ্যকবার এক্সিকিউট করার জন্য for লুপ ব্যবহৃত হয়।

৭। কাউন্টার ভেরিয়েবল বলতে কী বুঝায়?

উত্তর ঃ সাধারণত for লুপের আবর্তন সংখ্যা গণনা করার জন্য ব্যবহৃত ভেরিয়েবলকে কাউন্টার ভেরিয়েবল বলে।

৮। for লুপের সিনট্যাক্স লেখ।

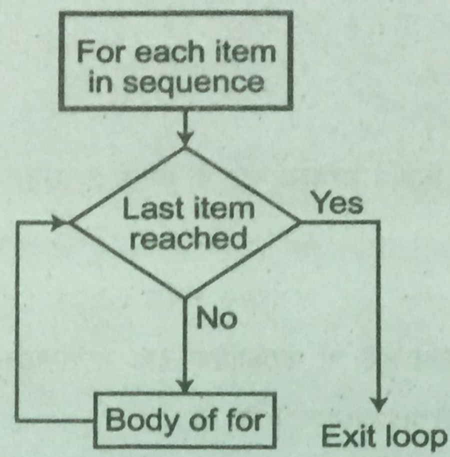
উত্তর ঃ for লুপের সিনট্যাক্স হলো ঃ

```
for <Variable> in range(<Number of Times>):
<Statements>
```

উল্লেখ্য যে, for লুপ-এর <Statements> অংশ অবশ্যই ইনডেন্টেশন দিতে হবে।

৯। **for লুপের ফ্লোচার্ট উল্লেখ কর।**

উত্তর ঃ **For loop-এর ফ্লো-চার্ট (Flowchart of for loop) ঃ**



চিত্র ঃ Flowchart of for loop

১০। **range() ফাংশনে কয় ধরনের ও কী কী প্যারামিটার আছে?**

উত্তর ঃ range() Function -এ দুই সেট প্যারামিটার আছে, যথা–

➢ range(Stop)
➢ range(Start, Stop, Step)

এখানে,

➢ Start হলো সিকুয়েন্সের ইনিশিয়াল ভ্যালু (শুরুর মান)
➢ Stop হলো সর্বশেষ মানের পূর্বের মান এবং
➢ Step হলো কাউন্টার ভেরিয়েবলের ইনক্রিমেন্ট বা ডিক্রিমেন্ট (অর্থাৎ মান কত করে বাড়বে বা কমবে) ভ্যালুকে বুঝায়।

১১। **range() ফাংশনের বিবেচ্য বিষয়সমূহ কী কী?**

উত্তর ঃ range() ফাংশনের বিবেচ্য বিষয়সমূহ হচ্ছে ঃ

➢ range() ফাংশনের সকল প্যারামিটার অবশ্যই ইন্টিজার (পূর্ণ সংখ্যা) হতে হবে।
➢ range() ফাংশনের প্যারামিটারসমূহ পজিটিভ বা নেগেটিভ যে-কোনো সংখ্যা হতে পারে।
➢ পাইথনে range() ফাংশনের ইনডেক্স সাধারণত ০ থেকে শুরু হয়।

১২। **নিম্নবর্ণিত range() ফাংশনসমূহের আউটপুট কী হবে?**

range(10), range(1, 10), range(1, 10, 2), range(10, 0, –2) , range(–5, 5)

উত্তর ঃ

range() ফাংশন	**আউটপুট**
range(10)	0,1,2,3,4,5,6,7,8,9
range(1, 10)	1,2,3,4,5,6,7,8,9
range(1, 10, 2)	1,3,5,7,9
range(10, 0, –2)	10,8,6,4,2
range((5, 5)	–5,–4,–3,–2,–1,0,1,2,3,4

১৩। **While লুপ ব্যবহার করা হয় কেন?**

উত্তর : একই কোড বার বার রান করানোর জন্য while লুপ ব্যবহার করা হয়। এতে একটি কন্ডিশন ব্যবহার করা হয়, যদি কন্ডিশনটি সত্য হয়, তাহলে while লুপের ভেতরে থাকা কোডগুলো রান হবে। আর যদি কন্ডিশনটি মিথ্যা হয়, তাহলে while লুপের ভেতরের কোডগুলো রান হবে না।

১৪। **while লুপের সিনট্যাক্স লেখ।**

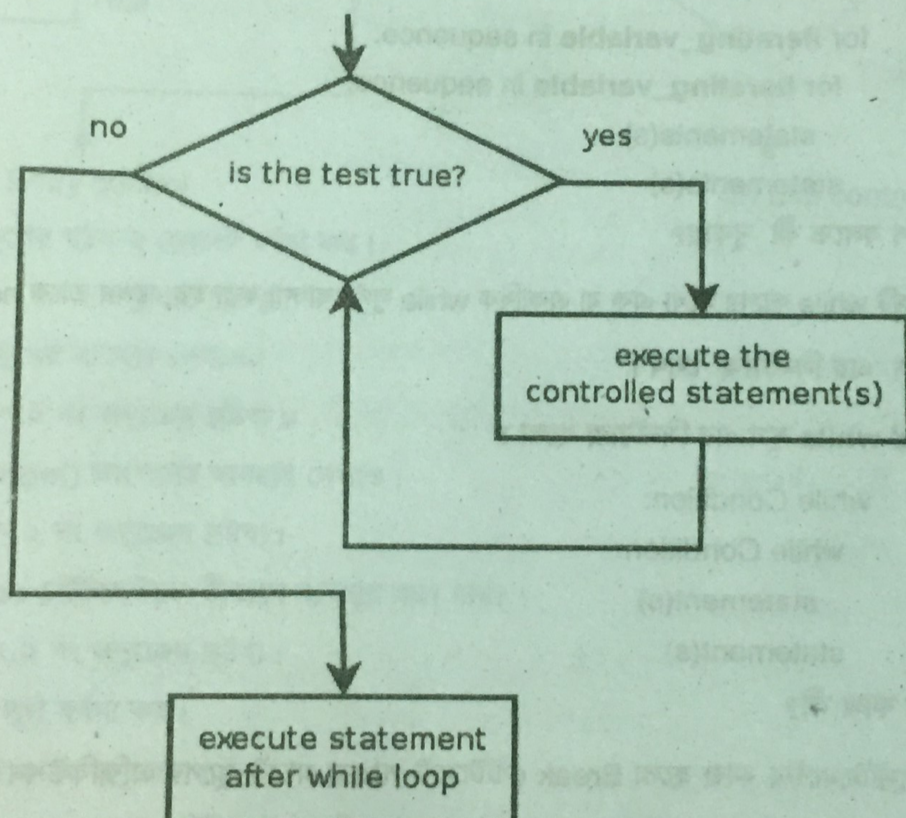
উত্তর : while লুপের সাধারণ গঠন হচ্ছে ঃ

```
while Condition :
    Statements
```

while লুপের কন্ডিশনটি যতক্ষণ পর্যন্ত সত্য হবে, ততক্ষক্ষ পর্যন্ত while লুপটি চলবে। কন্ডিশন মিথ্যা হলেই প্রোগ্রামের এক্সিকিউশন শেষ হয়ে যাবে।

১৫। **while লুপের ফ্লোচার্ট উল্লেখ কর।**

উত্তর : while লুপের ফ্লোচার্ট হলো ঃ



১৬। **ইনফিনিট লুপিং বলতে কী বুঝায়?**

উত্তর : প্রোগ্রামের এক বা একাধিক স্টেটমেন্টকে শর্তহীনভাবে বা অনির্দিষ্ট সংখ্যকবার আবর্তন করানোর প্রক্রিয়াকে ইনফিনিট (**Infinite**) লুপিং বলে।

১৭। **পাইথনে ইনফিনিট লুপিং কত প্রকার ও কী কী ?**

উত্তর : পাইথনে দুই ধরনের ইনফিনিট লুপ তৈরি করা যায়, যথা–

- Infinite for **লুপ** (Infinite for loop)
- Infinite while **লুপ** (Infinite while **লুপ** loop)

১৮। nested লুপিং বলতে কী বুঝায়?

উত্তর ঃ একটি লুপিং স্টেটমেন্টের মধ্যে অন্য কোনো এক বা একাধিক লুপিং স্টেটমেন্ট ব্যবহার করলে তাকে নেস্টেড লুপিং বলে। এক্ষত্রে মধ্যবর্তী লুপকে ইনার লুপ এবং বহিস্থ লুপকে আউটার লুপ বলে। আউটার লুপের পূর্বে ইনার লুপের কাজ শেষ হয়।

১৯। nested for লুপ বলতে কী বুঝায়?

উত্তর ঃ যখন একটি for লুপ স্টেটমেন্টের মধ্যে অন্য কোনো এক বা একাধিক for লুপ স্টেটমেন্ট ব্যবহার করা হয় তখন তাকে নেস্টেড for লুপ বলে। লক্ষণীয় যে, আউটার লুপ কন্ডিশন সত্য হলে ইনার লুপের স্টেটমেন্ট এক্সিকিউশন চলতে থাকে যতক্ষণ না ইনার লুপের কন্ডিশন মিথ্যা হবে। ইনার লুপের কন্ডিশন মিথ্যা হলে আবার আউটার লুপে যাবে এবং উপরোক্ত নিয়মে প্রক্রিয়াটি চলতে থাকবে যতক্ষণ না উভয় লুপই মিথ্যা হবে।

২০। nested for লুপ এর সিনট্যাক্স লেখ।

উত্তর ঃ **nested for লুপ এর সিনট্যাক্স হলো ঃ**

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

২১। nested while লুপ বলতে কী বুঝায়?

উত্তর ঃ যখন একটি while লুপের মধ্যে এক বা একাধিক while লুপ ঘোষণা করা হয়, তখন তাকে nested while লুপ বলে।

২২। nested while লুপ এর সিনট্যাক্স লেখ।

উত্তর ঃ **nested while লুপ এর সিনট্যাক্স হলো ঃ**

```
while Condition:
  while Condition:
    statement(s)
  statement(s)
```

২৩। break স্টেটমেন্টের কাজ কী?

উত্তর ঃ Break স্টেটমেন্টের কাজ হলো Break স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে লুপ থেকে বের হয়ে যাবে এবং লুপের বাইরে কোনো স্টেটমেন্ট থাকলে সেগুলোর এক্সিকিউশন শুরু করা।

২৪। continue স্টেটমেন্টের কাজ কী? [বাকাশিবো-২০১৭]

উত্তর ঃ Continue স্টেটমেন্টের কাজ হলো Continue স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে যাবে ঠিকই কিন্তু লুপ থেকে বের হবে না। শুধুমাত্র কন্ডিশনের অংশটুকু বাদ দেবে এবং প্রোগ্রামের বাকি অংশের কাজ সম্পন্ন করবে।

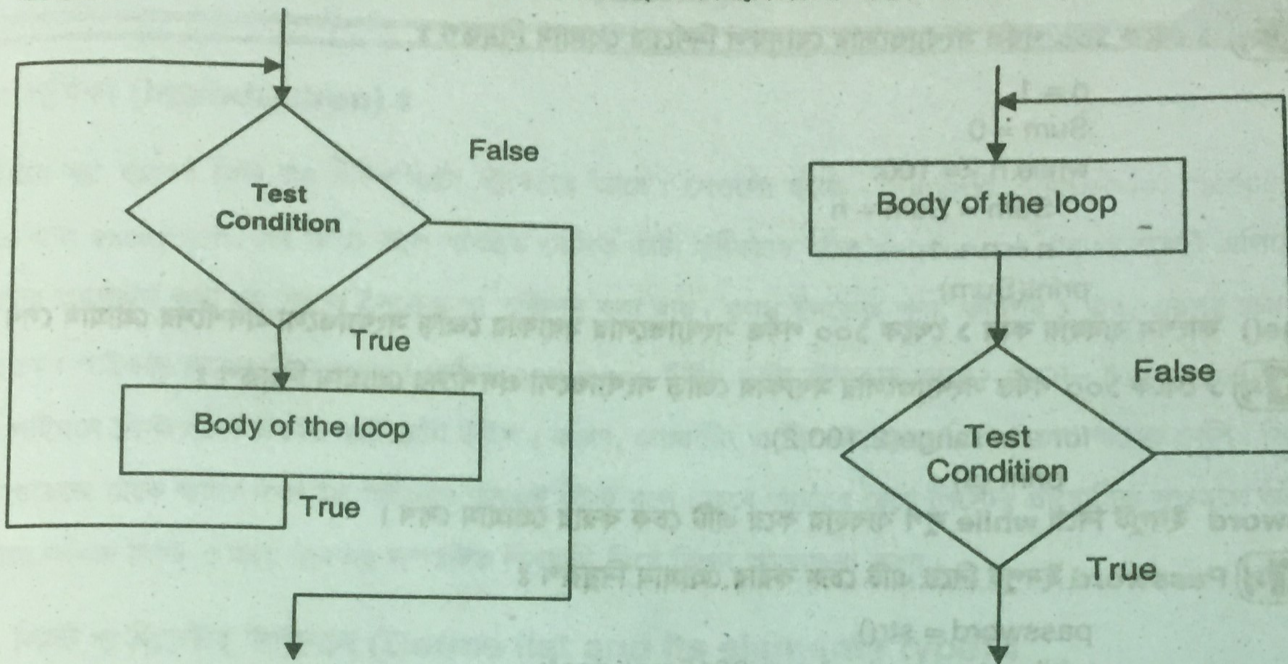
২৫। pass স্টেটমেন্ট ব্যবহার করা হয় কেন?

উত্তর ঃ Empty লুপ তৈরির জন্য Pass স্টেটমেন্ট ব্যবহৃত হয়। Pass স্টেটমেন্ট ব্যবহার করলে কোনো স্ট্রিং-এর সর্বশেষ ক্যারেক্টার বা লিস্ট এর সর্বশেষ সংখ্যা আউটপুট হিসেবে দেখাবে।

সংক্ষিপ্ত প্রশ্নোত্তর ঃ

১। **Entry controlled ও Exit controlled স্ট্রাকচার ফ্লোচার্টের মাধ্যমে দেখাও।**

উত্তর ঃ Entry controlled ও Exit controlled স্ট্রাকচার ফ্লোচার্টের মাধ্যমে দেখানো হলো ঃ



(a) Entry control

(b) Exit control

২। **উদাহরণসহ for লুপের গঠন ও ফ্লোচার্ট বর্ণনা কর।**

উত্তর সংকেত ঃ ৭.২ নং অনুচ্ছেদ দ্রষ্টব্য।

৩। **লিস্টের সঙ্গে for লুপের ব্যবহার দেখাও।**

উত্তর সংকেত ঃ ৭.২ নং অনুচ্ছেদ দ্রষ্টব্য।

৪। **for লুপের সঙ্গে range() ফাংশনের ব্যবহার দেখাও।**

উত্তর সংকেত ঃ ৭.২ নং অনুচ্ছেদ দ্রষ্টব্য।

৫। **for লুপের সঙ্গে else স্টেটমেন্টকে কীভাবে ব্যবহার করা যায়?**

উত্তর সংকেত ঃ ৭.২ নং অনুচ্ছেদ দ্রষ্টব্য।

৬। **উদাহরণসহ while লুপ বর্ণনা কর।**

উত্তর সংকেত ঃ ৭.৩ নং অনুচ্ছেদ দ্রষ্টব্য।

৭। **while লুপের সঙ্গে if else, elif স্টেটমেন্টকে কীভাবে ব্যবহার করা যায়? উদাহরণ দাও।**

উত্তর সংকেত ঃ ৭.৩ নং অনুচ্ছেদ দ্রষ্টব্য।

৮। **উদাহরণসহ nested for লুপের গঠন ও ফ্লোচার্ট বর্ণনা কর।**

উত্তর সংকেত ঃ ৭.৪ নং অনুচ্ছেদ দ্রষ্টব্য।

৯। **উদাহরণসহ nested while লুপের গঠন ও ফ্লোচার্ট বর্ণনা কর।**

উত্তর সংকেত ঃ ৭.৪ নং অনুচ্ছেদ দ্রষ্টব্য।

১০। **break ও continue স্টেটমেন্ট উদাহরণসহ বর্ণনা কর।**

উত্তর সংকেত ঃ ৭.৫ নং অনুচ্ছেদ দ্রষ্টব্য।

১১। Pass স্টেটমেন্টের ব্যবহার দেখাও।

উত্তর সংকেত ঃ ৭.৫ নং অনুচ্ছেদ দ্রষ্টব্য।

১২। while লুপ ব্যবহার করে ১ থেকে ১০০ পর্যন্ত সংখ্যাগুলোর যোগফল নির্ণয়ের প্রোগ্রাম লেখ।

উত্তর ঃ ১ থেকে ১০০ পর্যন্ত সংখ্যাগুলোর যোগফল নির্ণয়ের প্রোগ্রাম নিম্নরূপ ঃ

```
n = 1
Sum = 0
while n <= 100:
    Sum = Sum + n
    n = n + 1
print(Sum)
```

১৩। range() ফাংশন ব্যবহার করে ১ থেকে ১০০ পর্যন্ত সংখ্যাগুলোর মধ্যকার জোড় সংখ্যাগুলো প্রদর্শনের প্রোগ্রাম লেখ।

উত্তর ঃ ১ থেকে ১০০ পর্যন্ত সংখ্যাগুলোর মধ্যকার জোড় সংখ্যাগুলো প্রদর্শনের প্রোগ্রাম নিম্নরূপ ঃ

```
for a in range(2,100,2):
  print (a)
```

১৪। Password ইনপুট নিয়ে while লুপ ব্যবহার করে এটি চেক করার প্রোগ্রাম লেখ।

উত্তর ঃ Password ইনপুট নিয়ে এটি চেক করার প্রোগ্রাম নিম্নরূপ ঃ

```
password = str()
while password = "mahi2015mahdee":
password = input("Password: ")
print("Welcome, Please Come in")
```

১৫। For ও While লুপের মূল পার্থক্য কী? [বাকাশিবো-২০১৭]

উত্তর ঃ For লুপ ব্যবহার করা হয় যখন আমরা জানি ঠিক কতবার লুপ execute করা প্রয়োজন। অন্যদিকে While লুপ এমন পরিস্থিতিতে ব্যবহার করা হয় যখন আমরা জানি না ঠিক কতবার লুপ execute করা প্রয়োজন।

রচনামূলক প্রশ্নাবলি ঃ

১। বিভিন্ন প্রকার স্টেটমেন্টের সঙ্গে for লুপ স্টেটমেন্টের ব্যবহার উদাহরণসহ ব্যাখ্যা কর।

উত্তর সংকেত ঃ ৭.২ নং অনুচ্ছেদ দ্রষ্টব্য।

২। বিভিন্ন প্রকার স্টেটমেন্টের সঙ্গে while লুপ স্টেটমেন্টের ব্যবহার উদাহরণসহ ব্যাখ্যা কর।

উত্তর সংকেত ঃ ৭.৩ নং অনুচ্ছেদ দ্রষ্টব্য।

৩। while লুপ স্টেটমেন্ট ব্যবহার করে কতিপয় সংখ্যার নির্ণয়ের প্রোগ্রাম লেখ।

উত্তর সংকেত ঃ ৭.৩ নং অনুচ্ছেদ দ্রষ্টব্য।

৪। কোন একটি সংখ্যা ইনপুট নিয়ে এটির নামতা প্রিন্ট করার প্রোগ্রাম লেখ।

উত্তর সংকেত ঃ ৭.৬ নং অনুচ্ছেদ দ্রষ্টব্য।