# Conditional and Looping Constructs

## Very Short Answer type Questions [1 mark each]

**Question 1:**
Why do we use 'break' statement ?
**Answer:**
The 'break' statement can be used to terminate the loop.

**Question 2:**
What gets printed with the following code ?

```
x = True
y = False
z = False
if x or y and z :
print "yes"
else:
print "no
```

**Answer:**
Yes

**Question 3:**
What gets printed with the following code ?

```
x = True
y = False
z = False
if not x or y :
print 1
elif not x or not y and z:
print 2
elif not x or y or not y and x:
print 3
else:
print 4
```

**Answer:**
3

**Question 4:**
What gets printed with given code ?

```
f = None
for i in range (5):
```

with open(“data.txt”, “w”) as f:
if i > 2:
break
print f.closed

**Answer:**

True

**Question 5:**

Which numbers are printed?

```
for i in range(2):
print i
for i in range(4,6):
print i
```

**Answer:**

0,1,4, 5

**Question 6:**

What gets printed?

```
import re
sum = 0
pattern = ‘back’
if re.match(pattern, ‘backup.txt’):
sum + = 1
if re.match(pattern, ‘text.back’):
sum + = 2
if re.search(pattern, ‘backup.txt’):
sum + = 4
if re.search(pattern, ‘text.back’):
sum + = 8
print sum
```

**Answer:**

13

**Question 7:**

Write the syntax of an ’if statement1 in Python programming language

**Answer:**

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

**Question 8:**

Write the syntax of an if…..else statement in Python programming language

**Answer:**
if expression:
statement(s)
else:
statement(s)

**Question 9:**
Write the output
#!/usr/bin/python
var =100
if(var==100):
print “Value of expression is 100”
print “Good bye!
**Answer:**
Value of expression is 100
Good bye

**Question 10:**
Define while loops.
**Answer:**
A while loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

**Question 11:**
Write the syntax of a while loop.
**Answer:**
The syntax of a while loop in Python programming language is :
while expression :
statement(s)

**Question 12:**
What happened when the condition of while loop becomes false ?
**Answer:**
When the condition becomes false, program control passes to the line immediately following the loop.

**Question 13:**
Write the syntax of a for loop.
**Answer:**
The syntax of a for loop look is as follows:
for iterating_var in sequence :
statements(s)

**Question 14:**
What do you mean by “continue statement” ?
**Answer:**
It causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

**Question 15:**
What do you mean by “pass statement” ?
**Answer:**
The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

**Question 16:**
Write the syntax of “break” statement.
**Answer:**
The syntax for a break statement in Python is as follows : break

**Question 17:**
Does Python support “switch” statements?
**Answer:**
No, Python does not currently support switch or case statements as in other languages.

## Short Answer type Questions [2 mark each]

**Question 1:**
What is a statement ? What is the significance of an empty statement ?
**Answer:**
A statement is an instruction given to the computer to perform any kind of action.
An empty statement is useful in situations where the code requires a statement but does not require logic. To fill these two requirements simultaneously, empty statement is used. Python offers ‘pass’ statement as an empty statement.

**Question 2:**
What is the difference between determinable loop and non-determinable loop ?
**Answer:**
The ‘for loop’ can be labelled as ‘determinable
Ans. Value of expression is 100 Good bye!
loop’ as number of its iterations can be determined before-hand as the size of the sequence, it is operating upon.
The ‘while loop’ can be ‘non-determined loop’ as its number of iterations cannot be determined before-hand. Its iterations depend upon the result of a test-condition, which cannot be determined before-hand.

**Question 3:**
What are the two types of else clause in Python ? Ans. The two types of Python else clauses are :
**Answer:**
(i) else in an if statement
(ii) else in a loop statement
The else clause of an if statement is executed when the condition of the if statement results into false.
The else clause of a loop is executed when the loop is terminating normally i.e. when its test-condition has gone false for a while loop or when the for loop has executed the last value in sequence.

**Question 4:**
Explain nested if…. else.
**Answer:**
There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if construct. In a nested if construct, you can have an if…elif…else construct inside another if…elif… else construct

**Question 5:**
Write the syntax of a for loop and also given an example.
**Answer:**
The syntax of a for loop looks as follows : for iterating_var in sequence: statements(s)
Example : for i in range (4):
print i
output : 0 1 2 3

**Question 6:**
What do you mean by “for loop” ?
**Answer:**
A for loop is à Python statement which repeats a group of statements a specified number of times. You can use any object (such as strings, arrays, lists, tuples, diet and so on) in a for loop in Python.

**Question 7:**
Explain If…else statements
**Answer:**
If…else statements
An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a false value.
The else statement is an optional statement and there could be at most only one else

statement following if
Syntax:
The syntax of the if…else statement is :
if expression :
statement(s)
else:
statement(s)

**Question 8:**
What do you mean by decision making ?
**Answer:**
Structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be- true, and optionally, other statements to be executed if the condition is determined to be false.

**Question 9:**
Create a function addNumbers(x) that takes a number as an argument and adds all the integers between 1 and the number (inclusive) and returns the total number.
**Answer:**
```
def add Numbers (num):
total = 0
i = 1
while i< = num:
total + = i
i+ = 1
return total
```

**Question 10:**
Create a function addNumbers(start, end) that adds all the integers between the start and end value (inclusive) and returns the total sum.
**Answer:**
```
def addNumbers(start, end):
total = 0
i = start
while start < = end:
total + = start
start + = 1
return total
```

**Question 11:**

Create a function countPages(x) that takes the number of pages of a book as an argument and counts the number of times the digit '1' appears in the page number.

**Answer:**

```
def countPages(num):
total = 0
i = 1
while i<=num:
page_no = str(i)
total + = page_no.count('1')
i+ = l
return total
```

**Question 12:**

Which of the following does not produce the same output?

```
cnt = 0
while True:
    print cnt
    cnt+=1
    if cnt&gt;5:
        break
```

```
cnt = 0
while cnt &lt;6:
    while cnt &lt;6:
        print cnt
        cnt+=1
```

```
cnt = 0
while cnt &lt;6:
   print cnt
   cnt+=1
```

```
cnt = 0
while cnt &lt;6:
   print cnt
   if cnt==6:
      continue
   cnt+=1
```

**Answer:**
All give the same output.

**Question 13:**
Create a function factorial(x) that takes an integer and returns the product of all the positive integers less than or equal to n.

**Answer:**
```
def factorial(num):
product =1
i = num
while i > 0:
product *=i
i-=l
return product
```

**Question 14:**
Create a function doubleFactorial(n) that takes an odd integer and returns the product of all odd values upto the value n (where n=2k-l).

**Answer:**
```
def doubleFactorial(num):
product = 1
i = 0
k = 0
while k <num:
k = 2*i+l
product * = k
```

```
i += 1
return product
```

**Question 15:**

Create a function that computes the approximation of pi, based on the number of iterations specified, pi can be computed by 4*(l-l/3+l/5-l/7+l/9-…).

**Answer:**

```
def piApprox(num):
i = 1
pi = 0
while i<=num:
#set ‘while’ termination condition
pi +=((4/float(2*i-l)n-l)**(i+l))
#compute the ith term of the series
i+=l    # update i
return pi
```

**Question 16:**

What gets printed?

```
country_counter = {}
def addone(country):
if country in country_counter:
country_counter[country] + = 1
else:
country_counter[country] = 1
addone(‘China’)
addone(‘Japan’)
addone(‘china’)
print len(country_counter)
```

**Answer:**

3

**Question 17:**

What gets printed?

```
namesl = [‘Amir’, ‘Barry’, ‘Chales’, ‘Dao’]
if ‘amir’ in namesl:
print 1
else:
print 2
```

**Answer:**

2

**Question 18:**
What gets printed?

```
x = 0
y = l
a = cmp(x,y)
if a < x:
print “a”
elif a = = x:
print “b”
else:
print “c”
```

**Answer:**
a

**Question 19:**
What gets printed ?

```
x = 1
y = “2”
z = 3
sum = 0
for i in (x,y,z):
if isinstance(i, int):
sum + = i
print sum
```

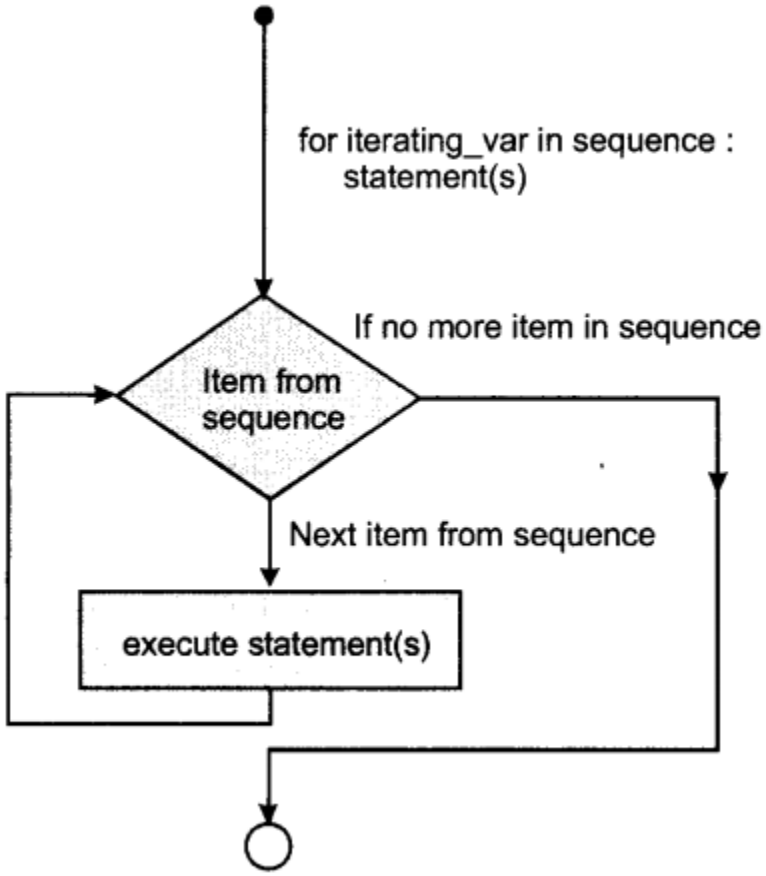
**Answer:**
4

## Long Answer type Questions [4 mark each]

**Question 1:**
Explain Flow Diagram of “for loop”

**Answer:**
If a sequence contains an expression list, it is evaluated first. Then, the first item in the sequence is assigned to the iterating variable iterating_var. Next, the statements bloc is executed. Each item in the list is assigned t iterating_var, and the statements(s) block is

executed until the entire sequence is exhausted.




**Question 2:**

Write an example to illustrates the combination of an else statement with a for statement

**Answer:**

The following example illustrates the combination of an else statement with a for statement that searches for prime numbers from 10 through 20.

```
# !/usr/bin/python
for num in range(10,20):
#to iterate between 10 to 20
for i in range(2,num):
#to iterate on the factors of the number
if num%i == 0:
#to determine the first factor
j=num/i
#to calculate the second factor
print '%d equals %d * %d' % (num,i,j)
break
#to move to the next number, the
#first for
else:
```

```
# else part of the loop
print num, 'is a prime number'
```

**When the above code is executed, it produces following result :**

```
10 equals 2*5
11 is a prime number
12 equals 2*6
13 is a prime number
14 equals 2*7
15 equals 3*5
16 equals 2*8
17 is a prime number
18 equals 2*9
19 is a prime number
```

**Question 3:**

Write a program to explain "for loop"

**Answer:**

```
#!/usr/bin/python
for letter in 'Python':    # First Example
print 'Current Letter :', letter
fruits = ['banana', 'apple', 'mango']
for fruit in fruits:    # Second Example
print 'Current fruit :', fruit
print "Good bye!"
```

**When the above code is executed, it produces following result:**

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

**Question 4:**

Create a function that takes in a positive integer and return a list of prime numbers. A prime number is only divisible by 1 and itself.

**Answer:**

```
def primeNumbers(num):
primes = [ ]
i = 2
```

```
# iterates through range from 2 to
num(inclusive)
while i< =num:
# add ‘while’ condition
k = 2
is Prime = True
# check if prime number
while k<i: # add’while’condition
if i%k==0:
isPrime = False
k+ = l   # update k
if isPrime = =True:
primes.append(i)
i+ = l # update i
return primes
```

**Question 5:**

Create a function that takes in a positive number and return 2 integers such that the number is between the squares of the 2 integers. It returns the same integer twice if the number is a square of an integer.

**Answer:**

```
import math
def sqApprox(num):
i = 0
minsq = 0 # set lower bound
maxsq = math.ceil(num’l’num)
# set upper bound
while i< maxsq :
# set ‘while’ termination condition
if i*i< =num and i>minsq:
# complete inequality condition
minsq = i
if i*i> =num and i<maxsq:
# complete inequality condition
maxsq = i
i+=l
# update i so that ‘while’ will terminate
return (minsq, maxsq)
```

**Question 6:**

Write a function estimatePi( ) to estimate and return the value of pi based on the formula found by an Indian Mathematician Srinivasa Ramanujan. It should use a while loop to compute the terms of the summation until the last term is smaller than le-15. The formula for estimating

pi is given below :

$$\frac{1}{\pi} = \frac{2\pi}{9801} \sum_{k=0}^{\infty} \frac{(4K)!(1103+26390*K)}{(k)!369^{4*k}}$$

where k! is the factorial of k.

**Answer:**

```
import math
def factorial(n):
if n = = 0:
return 1
else:
return n * factorial(n-l)


def series_term(k):
a=factorial(4*k)
b=(1103+26390*k)
c=factoriall(k)
d=c**4
e=396**(4*k)
return float(a'fb)/(d'l'e)


def estimatePi():
k=0
final=0
while True:
term = series_term(k)
final + = term
k+=l
if term < 1.0e-15:
break
f=2*math.sqrt(2)/9801
pi = 1.0/(final * f)
return pi
```

**Question 7:**

Given a positive integer, write a function that computes the prime factors that can be multplied together to get back the same integer.

**Answer:**

```
def primeFactorization(num):
factors = [ ]
lastresult = num
# 1 is a special case
if num == 1:
```

```
return [ ]
while True:
if lastresult == 1:
break
c = 2
while True:
if lastresult % c = = 0:
break
c += 1
factors.append(c)
lastresult/= c
return factors
```

**Question 8:**

The smallest common multiple of two or more numbers is called the lowest common multiple (LCM). Given a list of integers, find the lowest common multiple.

**Answer:**

```
def LCM(nums):
nums.sort()
biggest=nums[-l]
multiplier=1
while sum([(multiplier'|,biggest)%num for num in nums])!=0:
multiplier + = 1
return biggest*multiplier
```

**Question 9:**

Write a program to explain "if statement"

**Answer:**

```
# !/usr/bin/python
var1=100
if var1:
print "1-Got a true expression Value"
print var1
else:
print " 1-Got a false expression value"
print var1 var2=0
if var2 :
print "2-Got a true expression value"
print var2
else :
print "2-Got a false expression value"
print var2
```

print “Good bye!”

**When the above code is executed, it produces following result:**

1- Got a true expression value 100

2- Got a false expression value 0

Good bye!

**Question 10:**

Explain “elif statement” with example.

**Answer:**

elif statement : The elif statement allows you to check multiple expressions for truth value and execute a block of code as soon as one of the conditions evaluates to true. Like the else, the elif statement is optional. However, unlike else , for which there can be at most one statement, there can be an arbitrary number of elif statements following an if

The syntax of the if…elif statement is

If expression1 :

statement (s)

elif expression2 :

statement(s)

elif expression3 :

statement(s)

else:

statement(s)

**Example**

# !/usr/bin/py thon

var=100

if var = = 200 :

print “1-Got a true expression value”

print var

elif var==150:

print “2-Got a true expression value”

print var2

elif var ==100:

print “3-Got a true expression value”

print var

else:

print “4-Got a false expression value”

print var

print “Good bye!”

**when the above code is executed, it produces the following output :**

3- Got a true expression value

100

Good bye !

**Question 11:**
Explain nested if——else statement
**Answer:**
There may be a situation when you want to check for another condition after a condition resolves to true. In such a situation, you can use the nested if construct. In a nested if construct, you can have an if…elif…else construct inside another if…elif… else construct.
**Syntax:**
The syntax of the nested if…elif…else construct may be:
if expression1:
statement(s)
if expression2:
statement(s)
elif expression3:
statement(s)
else
statement(s)
elif expression4: statement(s) else:
statement(s)
Example:
# !/usr/bin/python if var < 200:
print “Expression value is less than 200”
if var==150:
print “=150”
elif var = =100:
print “=100”
elif var ==50:
print “=50”
elif var < 50 :
print “Expression value is less than 50”
else :
print “Could not find true expression”

**Question 12:**
Explain while loop in Python programming language.
**Answer:**
The syntax of a while loop in Python programming language is
while expression :
statement(s)
Here statement(s) may be a single statement or a block of statements. The condition may be any expression, and true is any nonzero value. The loop iterates while the condition is true. When the condition becomes false, program control passes to the line immediately following the loop. In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses

indentation as its method of grouping statements.

**Example**

```
# !/usr/bin/python
count=0
while (count < 9):
print 'The count is:',count
count=count +1
print "Good bye!"
```

OUTPUT

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
Good bye!
```

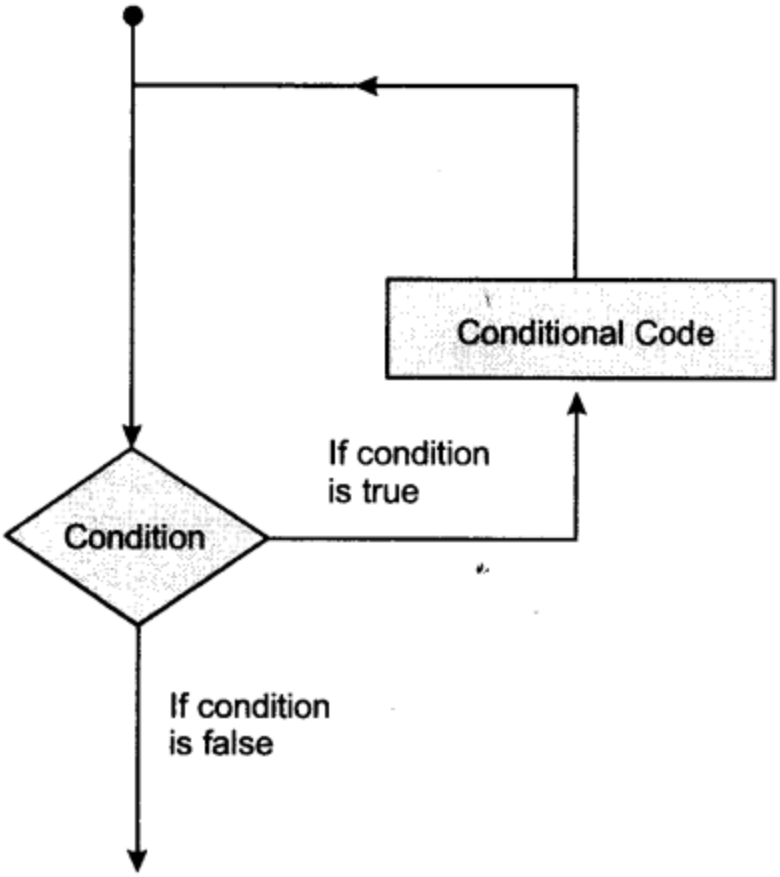
**Question 13:**

Explain loops in Python language with diagram.

**Answer:**

There may be a situation when you need to execute a block of code several number of times. In general statements are executed sequentially: The first statement in a function is executed first, followed by the second,and so on. Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and following is the general from of a loop statement in most of the programming languages




Python programming language provides
following types of loop to handle looping requirements.

**while loop :** Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

**for loop :** Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

**nested loops :** You can use one or more loop inside any another while, for or do..while loop.

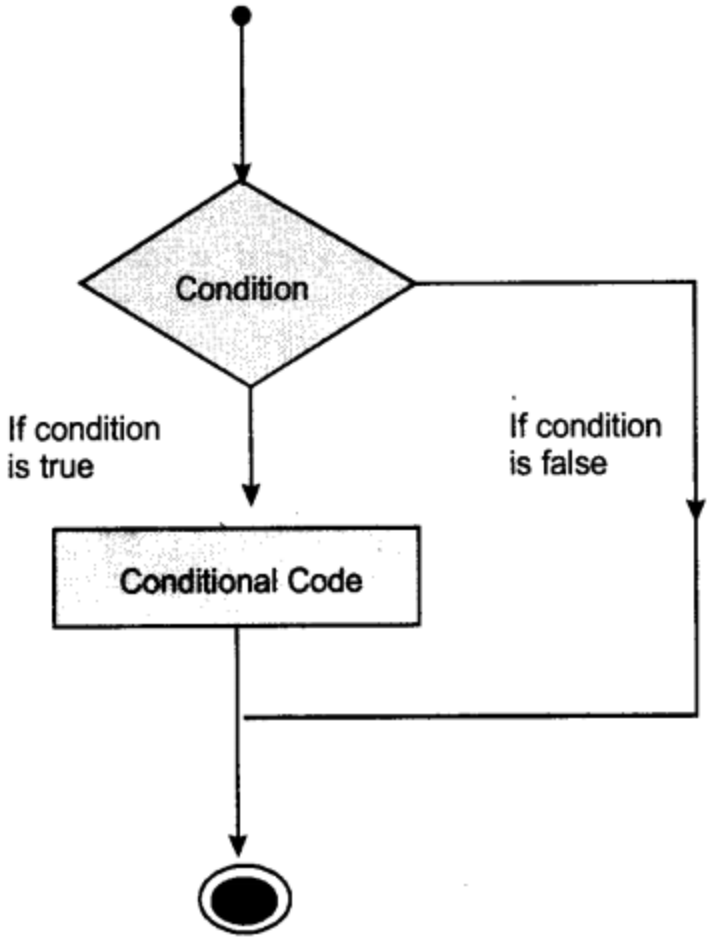
**Question 14:**

Explain Decision making structure with diagram.

**Answer:**

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general from of a typical decision making structure found in most of the programming languages :




Python programming language assumes any non-zero and non null values as true and if it is either zero or null then it is assumed as false value.
Python programming language provides follo-wing types of decision making statements.

| Statement | Description |
|---|---|
| if statements | An if statement consists of a boolean expression followed by one or more statements |
| if…else statements | An if statement can be followed by an optional else statement,which executes when the boolean expression is false |
| nested if statement | You can use one if or else if statement inside another if or else if statement(s) |

**Question 15:**
Write a function to print the Fibonacci Series upto input limit
**Answer:**

```
def fibonacci (num)
if (num = = 0):
return 0
elif (num = = 1):
return 1
else :
```

```
return fibonacci (num-1) + fibonacci
(num-2)
cal = input (“Enter the input limit”)
print fibonacci (val)
```

**Question 16:**

Print all multiples of 13 that are smaller than 100. Use the range function in the following manner : range (start, end, step) where ‘Start’ is the starting value of the counter, ‘end’ is the end value and ‘step’ is the amount by which the counter is increased each time.

**Answer:**

```
# program to print multiples of 13 smaller than 100 for counter in range (13,100,13) :
print counter
```

**Question 17:**

Write a program using while loop that asks the user for a number, and prints a countdown from that number to zero.

**Answer:**

```
num = input (“Enter the number”)
while (num > = 1) :
print (“num = “, num)
num – num-1
print (“Finish”)
```

**Question 18:**

Using random module, Simulate tossing a coin N times.

**Answer:**

```
N = input (“Number of tossing a coin”) from random import choice
heads = 0
tails = 0
coin = [‘head’, ‘tail’]
for n in range (N) :
toss = choice (coin)
if toss = ‘head’ :
heads = heads + 1
if toss = = ‘tail’ :
tails = tails + 1
print (‘Heads:’, heads)
print (‘Tails:’, tails)
```

**Question 19:**

Write a program using a for loop, that calculates exponentials. Your program should ask for base and exp. value from user.

**Answer:**

```
x = input (“Enter a base number”)
y = input (“Enter a exp. number”)
if y > 0 :
for x in range (y) :
x = x * x
else
for x in range (y) :
x = 1/x * x
print x
```

**Question 20:**

Following code is meant to be an interactive grade calculation script that converts from a percentage into a letter grade :
90 and above is A+,
80-90 is A,
60-80 is A-,
and everything below is fail.
Write the program to display grade of a user ?

**Answer:**

```
grade = input (“Enter the grade percentage”) if grade > = 90 :
print “Grade is A+” if 80 < = grade < 90 : print “Grade is A” elsif 60 < = grade : print “Grade is A-”
else grade < 60 :
print “You are fail”
```

**Question 21:**

Write a Python script to read an integer > 1000 and reverse the number.

**Answer:**

```
num = int (raw_input(“Enter an integer > 1000 : “))
tnum = num
reverse = 0
while tnum :
digit = tnum %10
tnum = tnum /10
reverse = reverse * 10 + digit
print “Reverse of”, num, “is”, reverse
```

**Question 22:**

Write a Python script to generate divisors of a number.

**Answer:**

```
num = int (raw_input (“Enter an integer:”))
mid = num/2
print “The divisors of”, num, “are:”
```

```
for a in range (2, mid +1):
if num %a = = 0:
print a,
else:
print”End”
```

**Question 23:**

Write a Python script that display first ten Mersenne numbers. Mersenne number are in the form of 2n-l.

**Answer:**

```
def Mersenne (n):
return 2**n-1
print “First ten Mersenne numbers are :”
for a in range (1,11):
print Mersenne (a)
```

**Question 24:**

Write a Python script that display 20 Mersenne Prime Numbers ?

**Answer:**

```
def Mersenne (n):
return 2** n-1
def test prime (n):
mid = n/2 + 1
for t in range (2, mid):
if n%t = = 0:
return false
else :
return true
print “20 Mersenne Prime Numbers are :”
for a in range (1,21):
number = Mersenne (a)
prime = test prime (number)
if prime :
print number, “Prime”
else :
print number
```

**Question 25:**

Write a Python script to find sum of the series :

s = 1 + x + x2 + x3 + …. + xn

**Answer:**

```
x = float (raw_input (“Enter value of x :”)
n = int (raw_input (“Enter value of n :”))
```

```
s = 0
for a in range (n + 1) :
s + = x**a
print “Sum of series”, s
```

**Question 26:**

Write Python script to print the following pattern:

```
1
1  3
1  3  5
1  3  5  7
```

**Answer:**

```
for a in range (3,10,2) :
for b in range (1, a, 2):
print b,
print
```

**Question 27:**

Write Python script to print the following pattern

```
1
1  1
1  1  1
1  1  1  1
1  1  1  1  1
```

**Answer:**

```
n = 1
for a in range (5) :
print n
n = n *10 + 1
```

**Question 28:**

Write Python script to calculate sum of following series :

s = (1) + (1+2) + (1+2+3) +…+ (1+2+3+…+n)

**Answer:**

```
sum 0
n = int (raw_input (“Enter the value of n :”)]
for a in range (2, n+2):
term = 0
for b in range (1,1):
```

```
term + = b
print “Term”, (a-1), “:”,term
sum + = term
print “Sum of”, n, “term is”, sum
```

**Question 29:**

Write a Python program to check if a given number is an Armstrong number.

**Answer:**

```
sum = 0
n = int (raw_input (“Enter number”)
n1 = n
while (n > 0):
a = n%10
sum = sum + (a * a * a)
n = n/10
if (sum ==n1):
print n, “is an Armstrong number”
else:
print n, “is not an Armstrong number”
```