

## 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 statement) 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 `Wusr/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 [4 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 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) (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 a 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 addNumbers(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 appears in the page number.

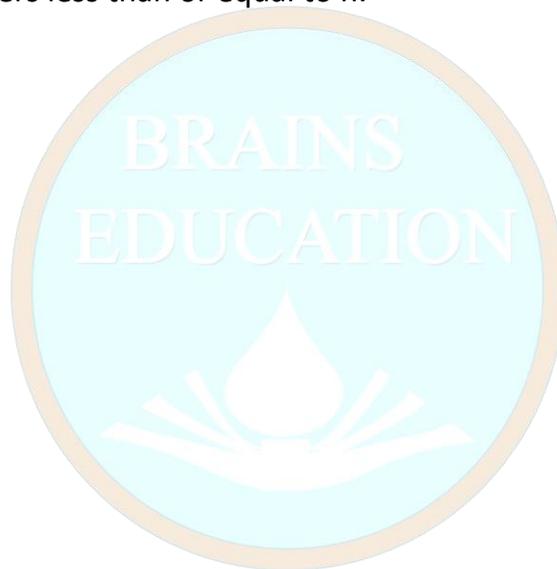
**Answer**

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

**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-=1  
    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  $n=2k-1$ ).

**Answer.**

```
def doubleFactorial(num):  
    product =1  
    i = 0
```

```
k = 0 while k < num:
```

```
k = 2*i+1
```

```
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  $4*(1-1/3+1/5-1/7+1/9-...)$ .

**Answer:**

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

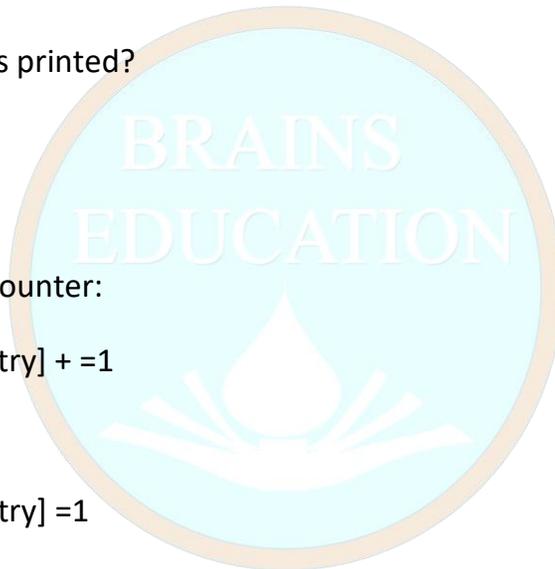
```
addone('china')
```

```
print len(country_counter)
```

**Answer 3**

**Question 17:** What gets printed?

```
namesl = ['Amir', 'Barry', 'chaes', 'Dao']
```



```
if 'amir' in names:
```

```
    print 1
```

```
else:
```

```
    print 2
```

**Answer 2**

**Question 18:** What gets printed?

```
x = 0
```

```
Y = 1
```

```
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"
```

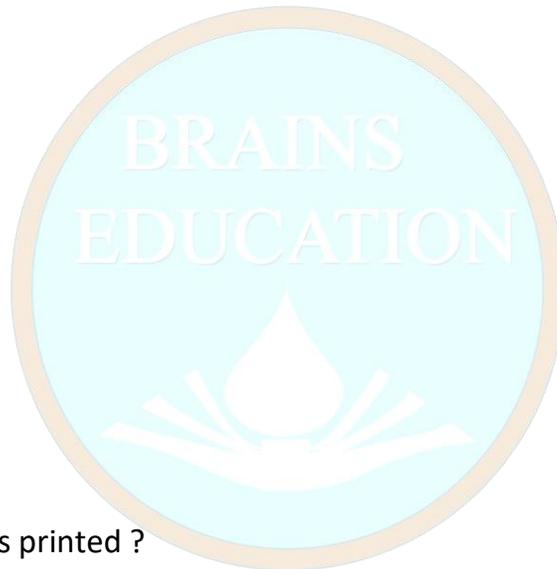
```
z = 3
```

```
sum = 0
```

```
for i in (x,y,z):
```

```
    if isinstance(i, int):
```

```
        sum += i
```



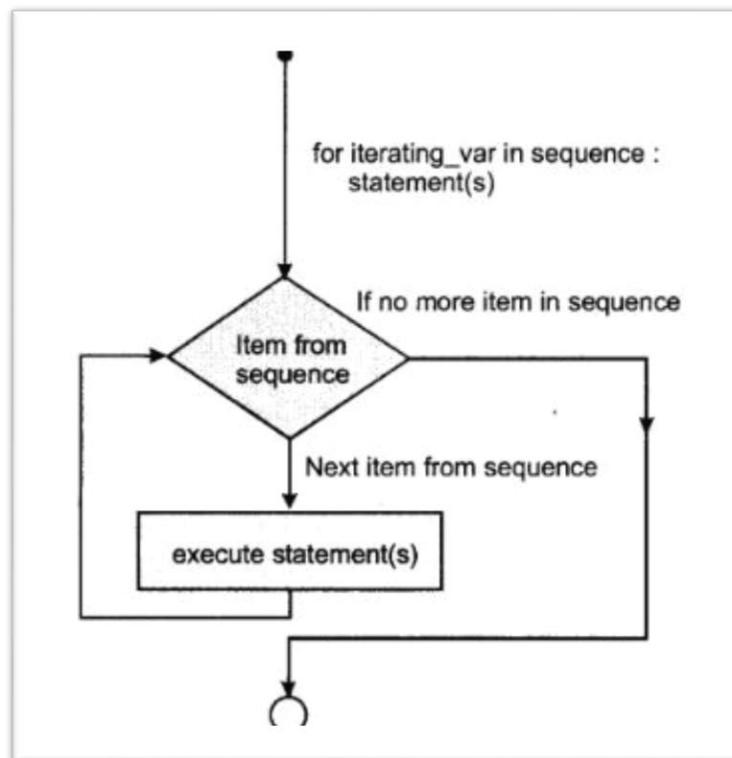
print sum

**Answer** 4

## Long Answer type Questions [10 marks 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. # Vusr/bin/python for num in range(10,20):

```
#!/usr/bin/python
```

```
#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 "s a orime 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', 'mangos']
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

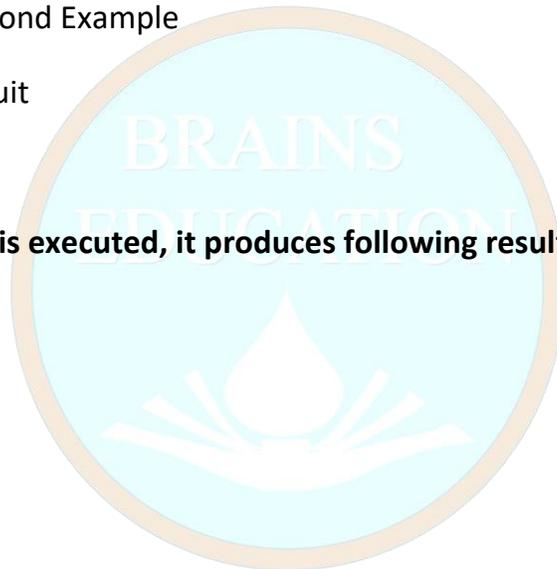
Current Letter : n

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**

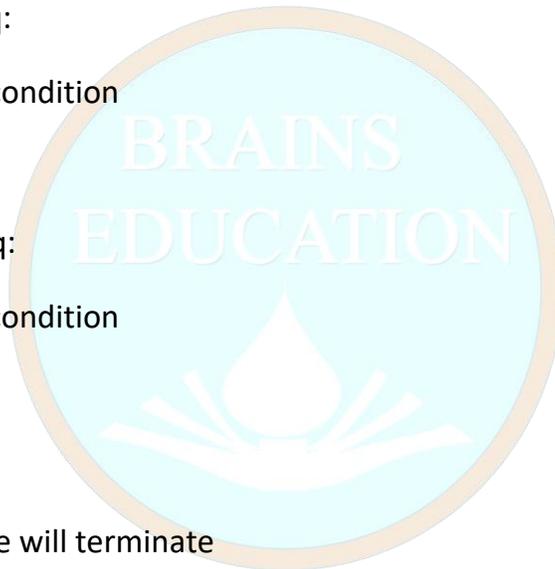
```
def primeNumbers(num):  
    primes = []  
    i = 2  
    # iterates through range from 2 to  
    num(inclusive)  
    while i <= num:  
        # add 'while condition'  
        k = 2  
        isPrime = True  
        # check if prime number while k < i: # add 'while' condition  
        if i % k == 0:  
            isPrime = False  
        k += 1 # update k  
        if isPrime == True:  
            primes.append(i)  
        i += 1 # 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**0.5) # 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 += 1  
    # update i so that 'while will terminate'  
    return (minsq, maxsq)
```



**Question 6:** Given a positive integer, write a function that computes the prime factors that can be multiplied 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: lastresult /= c factors.append(c) else: break c += 1 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):  
 biggest = nums[-1]  
 multiplier = 1  
 while sumd(multiplier\*biggest)%num for num in nums]!=0:  
 multiplier += 1  
 return biggest\*multiplier

**Question 9:** Write a program to explain 'if statement'

**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 expression) : statement (s)  
elif expression2 : statement(s)  
elif expression3 : statement(s)  
else: statement(s)

**Example** !/usr/bin/python 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 var  
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 !

