

[This question paper contains 12 printed pages.]

**Your Roll No.....**

**Sr. No. of Question Paper : 1546**

**G**

**Unique Paper Code : 2342011101**

**Name of the Paper : Programming using Python  
(DSC-1)**

**Name of the Course : B.Sc. (H) Computer Science**

**Semester : I**

**Duration : 3 Hours**

**Maximum Marks : 90**

**Instructions for Candidates**

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. **Section A** is compulsory.
3. Attempt **any 4** questions from **Section B**.
4. Parts of a question must be answered together.

**P.T.O.**

**SECTION A**  
**(Compulsory)**

1. (a) Give the pseudocode or flowchart for finding the largest of three numbers. (3)

- (b) Give type of error in the statements given below : (3)

`x, y, z = 20, '10', 0`

`d = [1, 2, 3]`

(i) `x / z`

(ii) `x + y`

(iii) `d[3]`

- (c) Determine the output of following code. State the values of **n** and **sum** for all iterations. (3)

`n, sum = 371, 0`

`while n > 0:`

`sum = sum + (n % 10) ** 3`

`n = n // 10`

`print(sum)`

- (d) Explain any two access modes available for file handling in python. Use suitable examples. (3)

- (e) Determine output of the following code snippet (3)

```
myStr = 'ihleD fo ytisrevinU'
myStr = myStr[::-1]
print(myStr + '-' + '110007')
print(len(myStr))
```

- (f) Write a function **studentData(n)**, that returns a dictionary with keys as the student roll number and value as student name. The function should accept roll number and name, for **n** students, as input from the user.

Example, the returned dictionary should look like :

{101: 'Sahil', 102: 'Sam',...} (5)

P.T.O.

- (g) In the given code snippet, state the value of **L1** after execution of each statement (3)

```
L1 = [10, 20, 30, 20]
```

```
L1.append(80)
```

```
print(L1)
```

```
L1.remove(20)
```

```
print(L1)
```

```
(L1.extend('aroma'))
```

```
print(L1)
```

- (h) Consider sets **a** and **b**: (3)

```
a = {5, 12, 33, 14, 55}
```

```
b = {101, 13, 14, 55, 16}
```

Find the value of **c** for each of the following statements :

(i) `c = a.union(b)`

(ii) `c = a & b`

(iii) `c = a.symmetric_difference(b)`

(i) Write a function **factors(n1)** that returns a set of all the factors of the number **n1**. (4)

### SECTION B

2. (a) Write a python program which contains the following functions : (7)

(i) **readfile(file1)** that prints the total number of lines present in file **file1**.

(ii) **copy(file1, file2)** that copies even lines of the file **file1** in file **file2**.

P.T.O.

(b) Evaluate the following expressions : (8)

(i)  $2 + 3 * 5 ** 2 \% 10 - 6$

(ii)  $1 > 2 < 3$

(iii)  $\sim (-6) == 5$

(iv)  $8 >> 3$

3. (a) Find the output of the following code : (5)

```
d = {'RED':4, 'GREEN': 14, 'BLUE':24}
```

```
dkeys = list(d.keys())
```

```
print(dkeys[0])
```

```
print('blue' in d)
```

```
d['ORANGE'] = 12
```

```
d['GREEN'] += 10
```

```
print(d)
```

(b) Write a function **process(str1)** that performs the following : (10)

(i) Calculates the frequency of each character in the string **str1**, using dictionary type. Print this dictionary.

(ii) Function should return the string which has the words in the reverse order.

Example: if **str1 = 'Best of luck Savita!'**, the function should return the string 'Savita! luck of Best'.

4. (a) State the value of **y** after each step: (7)

**x = 'quick sand'**

**y = x[3]**

**y += x[-4:-6:-1]**

**y += x[9] + x[-9]**

P.T.O.

```
y += x[-5] + x[-3] + x[3]
y += x[5] + x[2] + x[-2]
y = y.partition(' ')
y = y[0] + '@' + ' '.join(y[2].split(' '))
print(y)
```

(b) Describe the following string functions with examples : (8)

(i) capitalize()

(ii) isdigit()

(iii) upper()

(iv) isalpha()

5. (a) Consider the following set of statements : (7)

```
import copy
```

```
list1 = [ 1, 2, [ 3, 4]]
```



```
list2 = copy.copy(list1)
```

```
list3 = copy.deepcopy(list1)
```

```
list3[0] = 690
```

```
list1[2][0] = 75
```

Find and justify the values of list1, list2 and list3.

(b) Generate a list containing the cube of the odd numbers from 1 to n, using : (8)

(i) a user-defined function **myCube(n)**

(ii) list comprehension

6. (a) Determine the output of the following code: (7)

```
x, y = 4, 5
```

```
sum = 0
```

```
while (y > 0):
```

P.T.O.

```
if (y & 1):  
    sum = sum + x  
  
x = x << 1  
y = y >> 1  
  
print(x, y)  
  
print(sum)
```

(b) Write a program that does the following checks on the age entered by the user : (8)

(i) age should not contain alphabets or special characters

(ii) age should not be less than 21

Raise and handle appropriate exception(s).

7. (a) Compute the output of the following code : (7)

```
t1 = (1, 2, 3, 7, 9, 0, 5, 7)
```

```
t2 = (23, [24, 25])
```

```
print(max(t1))  
print(t1[1:3])  
print(t1.count(7))  
t2[1][0]= 5  
print(t2)  
t3 = t1 + tuple('India')  
print(t3)
```

- (b) Define a class **Drone** that contains following data members : (8)

Instance variables: **droneId** - id of drone

Class variable: **totalCount** - for keeping count of all the drones manufactured

The class should contain the following methods:

(i) **\_\_init\_\_()** - initialize data members and increment **totalCount**

(ii) **getId()** - returns **droneId**

P.T.O.

- (iii) **getTotalCount()** - returns **totalCount**
- (iv) **\_\_del\_\_()** - to destroy the object and decrement the **totalCount**