

1020

12

(b) Write a program to read a file myFile and perform the following operations: (10)

- (i) print the total number of lines in the file.
- (ii) copy even lines of the file to a file named evenFile and odd lines to another file named oddFile.

(500)

[This question paper contains 12 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 1020

D

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.

**SECTION A**

**(Compulsory)**

1. (a) What is an algorithm? Write an algorithm to solve the quadratic equation. (4)

P.T.O.

(b) Write a function `checkPrime(n)` to check whether the given number `n` is prime or not. The function should return 1 if the number `n` is prime else 0. Call this function in `main()` to check the number input by the user. (4)

(c) Evaluate the following expressions: (4)

(i) `9+3*2**2 != 9//4-2` and `'hello' >= 'Hello world'`

(ii) `20 ^ -22 & -5`

(d) Consider the dictionary `groupDict` representing student details of a group. (4)

`groupDict` is defined as follows :

```
groupDict = {'group': {
    'student': {'name': 'Mike',
               'marks': {'physics': 70, 'history': 80}
               },
             },
            'section': 'A'
}
```

Write the python code snippets for the following operations :

(b) Explain the type of exception raised in the following statements : (4)

(i) `x=0`  
`print(5/x)`

(ii) `print('The amount for the day is :'+ 300)`

(iii) `int('Morning')`

(iv) `L1 = [11,22,33,44,55]`  
`print(L1[5])`

(c) Describe the following methods for the class objects with suitable examples. (8)

(i) `__init__`

(ii) `__str__`

(iii) `__del__`

(iv) `__main__`

7. (a) Write a function `sumDigits(Num)` which computes the sum of digits of a number `Num` and returns it. (5)

- (iii) percentage ('200.0', 200.0)
- (b) Define the following functions to perform the operations on a string. (9)
- (i) countVowel(Str1) to count the vowels in a string.
  - (ii) replaceChar(Str1) to replace all occurrence of the character 'a' with a space.
  - (iii) reverseString(Str1) to reverse a string

Write a program to call these functions and provide the output for the given string Str1.

Str1 = 'Happiness depends upon ourselves!'

6. (a) Find the error(s) in the following code snippets: (3)
- (i) 

```
file1 = open('Myfile', 'w')
file1.read()
file1.close()
```
  - (ii) 

```
tuple1 = (2120, 'abc')
del tuple1[0]
```

- (i) Access the value of the subject **Chemistry**.
  - (ii) Extract the value of the key **group**, and use the default value as -1 if key is not found.
  - (iii) Create a copy of a **groupDict** into a dictionary object **newGroup**.
  - (iv) print the dictionary **newGroup** after removing the element with the key **section**.
- (e) Define a class Rectangle having the following structure: (6)
- Attributes: length, breadth
- Methods: `__init__()` for initializing the attributes.
- `getLength()` which returns the length of the rectangle.
- `perimeter()` which returns the perimeter of the rectangle.
- (f) Find the output for the following python scripts: (8)

```
(i) myString = 'Hello Everyone, Welcome to the session!'
    print(myString [len(myString) :: -1])
    print(myString[:-15] + myString[-15:])
    print(myString.partition('Welcome'))
    print(myString.rfind('to'))

(ii) X = ['Red', 'Blue', 'Green']
     Y = ['Yellow', 'White']
     X.extend(Y)
     X.append(23)
     X.pop()
     X.remove('Yellow')
     print(X)
     print(Y)

(iii) try:
        num = 8
        print(num + 'hello')
        print(num / 4)
    except ZeroDivisionError:
        print('Divided by zero')
    except (ValueError, TypeError):
        print('Error occurred')
    finally:
        print('Stop')
```

```
def percentage (marks, total):
    try:
        percent = (marks/total) * 100
    except ValueError:
        print('Value Error')
    except TypeError:
        print("Type Error")
    except ZeroDivisionError:
        print('Zero Division Error')
    except:
        print('Any other error')
    else:
        print(percent)
    finally:
        print('Completed!')
```

Explain the output corresponding to the following function calls.

(i) percentage (17.0, 20.0)

(ii) percentage (19.0, 0.0)

- (b) Consider two lists Lst1 and Lst2 declared as follows : (8)

```
Lst1 = ['green', 'blue']
```

```
Lst2 = ['blue', 'yellow']
```

Write a python script to do the following :

- (i) Convert list Lst1 as set **S1** and **Lst2** as set **S2**.
  - (ii) Add the elements of the list ['black', 'cyan'] to **S2**.
  - (iii) Find the symmetric difference between sets **S1** and **S2**.
  - (iv) Create a set **newSet** using set comprehension containing elements of **S1** with 's' added at the end of each element. The **newSet** should appear as:  
  
**newSet = {'blues', 'greens'}**
5. (a) Consider the **function percentage(marks, total)** that computes the percentage of marks for a student. (6)

- (iv) 

```
monthDays = {'Januray': 31, 'February': 28, 'March': 31}
Month = monthDays
Month['February'] += 1
print('monthDays before clear-->', monthDays)
print('Month before clear-->', Month)
Month.clear()
print('monthDays after clear-->', monthDays)
print('Month after clear-->', Month)
```

### SECTION B

2. (a) Write a program that accepts x and n as input to compute the following series. (6)

$$x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots \text{ n terms}$$

- (b) Consider the following code snippet (9)

```
for k in 'Computer Science':
```

```
    if k = 'e':
```

```
        S1
```

```
        print(k, end = ", ")
```

Compare the output when **S1** is replaced with each of the following statements:

(i) break

(ii) continue

(iii) pass

3. (a) Consider the following tuples (6)

T1 = (100, 200, 300)

T2 = ('Monday', 'Tuesday', 'Wednesday')

(i) Write a function **swapTuple(T1, T2)** to swap the values of T1 and T2.

The expected output is as follows:

T1 = ('Monday', 'Tuesday', 'Wednesday')

T2 = (100, 200, 300)

(ii) Write a function **mergeTuple(T1, T2)** to return a list of tuples containing the corresponding element from tuples T1 and T2. The expected output is as follows :

T3 = [('Monday', 100), ('Tuesday', 200), ('Wednesday', 300)]

(b) Consider list L1 as follows : (9)

L1 = [100, 200, 300, [400,500]]

Write code snippets to create the following lists :

(i) L2 as a shallow copy of L1

(ii) L3 as a deep copy of L1.

Demonstrate the effect of the following modifications in L1, L2 and L3:

(i) L1[2] = 900

(ii) L1[3][0] = 700

4. (a) Write a function **doubleDict()** that creates the dictionary **Dict1** where the keys are numbers between 1 and 5 and values are twice the keys. For example: if the key is 5, its value is 10. The function should return the dictionary **Dict1**.

Write a program that calls **doubleDict()** and prints the values of the following operations when applied to the **Dict1**.

(i) maximum key

(ii) sum of keys

(7)