

PYTHON

Practical of B.Sc.(Hons.) Computer Science (Semester 1)

**Prepared by : Ankit Kumar
Roll No. : 16005**

Question 1: WAP to find the roots of a quadratic equation.

CODE

```
#Question:WAP to find the roots of a quadratic equation.
import cmath
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
c=int(input("Enter the third number:"))
D=(b*b)-(4*a*c)
root_1=(-b+(cmath.sqrt(D)))/(2*a)
root_2=(-b-(cmath.sqrt(D)))/(2*a)
print('root_1=',root_1)
print('root_2=',root_2)
```

OUTPUT

```
Enter the first number:2
Enter the second number:4
Enter the third number:3
root_1= (-1+0.7071067811865476j)
root_2= (-1-0.7071067811865476j)
```

Question 2: Write a program to accept a number 'n' and

- A. Check if 'n' is prime
- B. Generate all prime numbers till 'n'
- C. Generate first 'n' prime numbers

CODE

```
#A
def prime1():
    n=int(input('Enter the number:'))
    flag=0
    if n<=1:
        print(n,'is not a prime number')
    elif n>2:
        for i in range(2,n):
            if n%i==0:
                flag=1
                break

        if flag==0:
            print(n,'is a prime number')
        else:
            print(n,'is not a prime number')

    elif n==2:
        print(n,'is a prime number')

prime1()

#B
def prime2():
    n=int(input('Enter the number:'))
    for a in range(2,n+1):
        if a>1:
            for j in range(2,a):
                if a%j==0:
                    break
            else:
                print(a)

prime2()

#C
def prime3():
    n = int(input("Enter the number : "))
    c = 2
    while n!=0:
        for i in range(2,c):
            if c%i==0:
                break
        else:
            print(c,end=" ")
            n-=1
        c+=1

prime3()
```

OUTPUT

```
Enter the number:5
5 is a prime number
Enter the number:7
2
3
5
7
Enter the number : 8
2 3 5 7 11 13 17 19
```

Question 3: WAP to create a pyramid of the character "*" and a reverse pyramid.

CODE

```
rows = int(input("Enter number of rows: "))

def pyramid(rows):
    for i in range(rows):
        for j in range(rows-i-1):
            print(' ',end='')
        for j in range(2*i+1):
            print('*',end='')
        print()
    print('\n')
    for i in range(rows):
        for j in range(i):
            print(' ',end='')
        for k in range(2*(rows-i)-1):
            print(" ",end='')
        print()

pyramid(rows)
```

OUTPUT

```
Enter number of rows: 5
```

```
  *
 ***
*****
*****
*****

*****
*****
 *****
  *****
   ***
    *
```

Question 4: WAP that accepts a character and perform the following:

- print whether the character is a letter or numeric digit or a special character.
- if the character is a letter, print whether the letter is uppercase or lowercase.
- if the character is a numeric digit, print its name in text(e.g. ,if input is 9,output is NINE)

CODE

```
'''a.finding data type of the character'''
msg=input("Enter the character:")
if msg.isalpha():
    print("Character is an alphabet.")
elif msg.isdigit():
    print("Character is a number.")
else:
    print("character is a special character.")

'''b.finding whether the character(if letter) is in uppercase or lowercase'''
msg=input("Enter the character:")
if msg.isalpha():
    if msg.islower():
        print("Character is in lowercase.")
    elif msg.isupper():
        print("character is in uppercase.")
else:
    print("character entered by you is not an alphabetic character.")

'''c.Printing in words,if the character entered is a numeric'''
NUMBER=input("Enter the number:")
if NUMBER.isdigit():
    if NUMBER=="0":
        print("ZERO")
    elif NUMBER=="1":
        print("ONE")
    elif NUMBER=="2":
        print("TWO")
    elif NUMBER=="3":
        print("THREE")
    elif NUMBER=="4":
        print("FOUR")
    elif NUMBER=="5":
        print("FIVE")
    elif NUMBER=="6":
        print("SIX")
    elif NUMBER=="7":
        print("SEVEN")
    elif NUMBER=="8":
        print("EIGHT")
    elif NUMBER=="9":
        print("NINE")
else:
    print("Enter a valid numeric character:")
```


OUTPUT

```
Enter the character:5
Character is a number.
Enter the character:7
character entered by you is not an alphabetic character.
Enter the number:3
THREE
```

Question 5: WAP to perform the following operations on a string:

- Find the frequency of a character in a string.
- Replace a character by another character in a string.
- Remove the first occurrence of a character from a string.
- Remove all occurrences of a character from a string.

CODE

```
'''finding frequency of a character in a string'''
msg_string=input("Enter the string:")
character=input("Enter the character to be count:")
frequency_ch=msg_string.count(character)
print("frequency of the character entered is:",frequency_ch)

'''replacing s character by another character in a string'''
msg_string=input("Enter the string:")
character=input("Enter the character to be replaced by new character:")
character1=input("Enter the character1 with which we want to replace character:")
msg_string=msg_string.replace(character,character1)
print(msg_string)

'''removing the first occurrence of a character from a string'''
msg_string=input("Enter the string:")
character=input("Enter the character whose first occurrence is to be removed:")
msg_string1=''
length=len(msg_string)
i=0
while(i<length):
    if(msg_string[i]==character):
        msg_string1=msg_string[0:i]+msg_string[i+1:length]
        break
    i=i+1
print("original string:",msg_string)
print("final string:",msg_string1)

'''removing all occurrence of a character in a string'''
msg=input("Enter the string:")
rem_char=input("Enter the character to be removed:")
print("The original string is:"+str(msg))
res=msg.replace(rem_char,"")
print("The string after character deletion:"+str(res))
```

OUTPUT

```
Enter the string: BALLOON
Enter the character to be count: O
frequency of the character entered is: 2
Enter the string: APPLE
Enter the character to be replaced by new character: E
Enter the character1 with which we want to replace character: A
APPLA
Enter the string: TIGER
Enter the character whose first occurrence is to be removed: R
original string: TIGER
final string: TIGE
Enter the string: MONKEY
Enter the character to be removed: K
The original string is: MONKEY
The string after character deletion: MONEY
```

Question 6: WAP to swap the first n character of two strings.

CODE

```
def swap_first_n_chars(string1, string2, n):
    string1_start = string1[:n]
    string2_start = string2[:n]
    string1 = string2_start + string1[n:]
    string2 = string1_start + string2[n:]
    return string1, string2

string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
n = int(input("Enter the value of n: "))
print(swap_first_n_chars(string1, string2, n))
```

OUTPUT

```
Enter the first string: TITAN
Enter the second string: GIGGLE
Enter the value of n: 3
('GIGAN', 'TITGLE')
```


Question 7: Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.

CODE

```
def check():
    #Taking input of string from user.
    string1=input("Enter first string:")
    string2=input("Enter second string:")
    list=[]
    #To check whether string2 is available in string1.
    if string2 in string1:
        for i in range(0,len(string1)):
            for j in range(0,len(string2)):
                if string1[i]==string2[j]:
                    if i not in list:
                        list.append(i)
                    continue
        print(list)
    else:
        print(-1)
check()
```

OUTPUT

```
Enter first string:PERMANENT
Enter second string:MANENT
[1, 3, 4, 5, 6, 7, 8]
```

Question 8: WAP to create a list of the cubes of only the even integers appearing in the input list

(may have elements of other types also) using the following:

- a. 'for' loop
- b. list comprehension'''

CODE

```
#Using for loop.
def num_list():
    num_list=eval(input("Enter your list:"))
    list_1=list(num_list)
    list_2=[]
    for i in list_1:
        if i%2==0:
            element=i**3
            list_2.append(element)
        else:
            None
    print(list_2)
num_list()
#Using list comprehension.num_list=eval(input("Enter your list:"))
num_list=eval(input("Enter your list:"))
new_list=[x**3 for x in num_list if x%2==0]
print(new_list)
```

OUTPUT

```
Enter your list: [1,2,3,4,5]
[8, 64]
Enter your list: [6,7,8,9]
[216, 512]
```

Question 9: WAP to read a file and

- m. Print the total number of characters, words and lines in the file.
- n. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.

- o. Print the words in reverse order.
- p. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.

CODE

```
def read_file(file):
    with open(file, "r") as f:
        content = f.read()
        # a. Print the total number of characters, words, and lines in the file
        num_chars = len(content)
        num_words = len(content.split())
        num_lines = content.count("\n")
        print("Total characters:", num_chars)
        print("Total words:", num_words)
        print("Total lines:", num_lines)

        # b. Calculate the frequency of each character in the file
        char_freq = {}
        for char in content:
            if char in char_freq:
                char_freq[char] += 1
            else:
                char_freq[char] = 1
        print("Character frequency:", char_freq)

        # c. Print the words in reverse order
        words = content.split()
        reversed_words = [word[::-1] for word in words]
        print("Reversed words:", reversed_words)

        # d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'
        lines = content.split("\n")
        with open("File1", "w") as f1:
            for i, line in enumerate(lines):
                if i % 2 == 0:
                    f1.write(line + "\n")
        with open("File2", "w") as f2:
            for i, line in enumerate(lines):
                if i % 2 == 1:
                    f2.write(line + "\n")

read_file("file.txt")

print()
```

TEXT FILE

```
file - Notepad
File Edit View

Discipline is a term of following certain rules and regulations on personal conduct to get successful in life.
The main purpose of discipline is to restrain people from acts of disobedience and negligence and to bring them under a rule.
Disciplined children are more dedicated to their studies and behave more responsible.
```

OUTPUT

```
Total characters: 323
Total words: 52
Total lines: 2
Character frequency: {'D': 2, 'i': 27, 's': 19, 'c': 13, 'p': 9, 'l': 15, 'n': 2, 'e': 37, ' ': 50, 'a': 15, 't': 15, 'r': 18, 'm': 6, 'o': 20, 'f': 7, 'w': 1, 'g': 6, 'u': 9, 'd': 15, '.': 3, '\n': 2, 'T': 1, 'h': 5, 'b': 4, 'v': 1}
Reversed words: ['enilpicsiD', 'si', 'a', 'mret', 'fo', 'gniwoollof', 'niatrec', 'selur', 'dna', 'snoitaluger', 'no', 'lanosrep', 'tcudnoc', 'ot', 'teg', 'lufssecus', 'ni', '.efil', 'ehT', 'niam', 'esoprup', 'fo', 'enilpicsid', 'si', 'ot', 'niartser', 'elpoep', 'morf', 'stca', 'fo', 'ecneidebosid', 'dna', 'ecnegilgen', 'dna', 'ot', 'gnirb', 'meht', 'rednu', 'a', '.elur', 'denilpicsiD', 'nerdlihc', 'era', 'erom', 'detacided', 'ot', 'rieht', 'seiduts', 'dna', 'evaheb', 'erom', '.elbisonpser']
```

FILE 1

```
File Edit View

Discipline is a term of following certain rules and regulations on personal conduct to get successful in life.
Disciplined children are more dedicated to their studies and behave more responsible.
```

FILE 2

```
File2 - Notepad
File Edit View

The main purpose of discipline is to restrain people from acts of disobedience and negligence and to bring them under a rule.
```

Question 10: WAP to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method

distance to calculate the distance between any two point objects.

CODE

```
import math

class Point:
    """
    A class representing a point in 2D space with x and y coordinates.
    """

    def __init__(self, x, y):
        """
        Initialize a new point with the given x and y coordinates.
        """
        self.x = x
        self.y = y

    def __repr__(self):
        """
        Return a string representation of the point.
        """
        return f'Point({self.x}, {self.y})'

    def distance(self, other):
        """
        Calculate the distance between this point and another point.
        """
        dx = self.x - other.x
        dy = self.y - other.y
        return math.sqrt(dx**2 + dy**2)

# Example usage:
p1 = Point(0, 0)
p2 = Point(3, 4)
```

OUTPUT

```
Point(0, 0)
Point(3, 4)
5.0
```

Question 11: Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.


```
def print_dict_of_cubes():
    """
    This function prints a dictionary where the keys are numbers between 1 and 5
    and the values are cubes of the keys.
    """
    # Create an empty dictionary
    cube_dict = {}

    # Iterate over the numbers between 1 and 5
    for i in range(1, 6):
        # Calculate the cube of the number
        cube = i**3
        # Add the number and its cube to the dictionary as a key-value pair
        cube_dict[i] = cube

    # Print the dictionary
    print(cube_dict)

# Call the function to print the dictionary
print_dict_of_cubes()
```

OUTPUT

```
{1: 1, 2: 8, 3: 27, 4: 64, 5: 125}
```

Question 12: Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:

- Print half the values of the tuple in one line and the other half in the next line.
- Print another tuple whose values are even numbers in the given tuple.
- Concatenate a tuple t2 = (11,13,15) with t1.
- Return maximum and minimum value from this tuple.

CODE

```
tuple_t1=(1,2,5,7,9,2,4,6,8,10)
#printing half the values of the tuple in one line and the other half in the next line.
tp1=tuple_t1[:5]
tp2=tuple_t1[5:]
print("First half values of tuple t1 is:",tp1)
print("Other half values of tuple t1 is:",tp2)
#printing another tuple whose values are even numbers in the given tuple.
list_1=list()
for i in tuple_t1:
    if i%2==0:
        list_1.append(i)
tuple_n=tuple(list_1)
print("Tuple whose values are even in tuple t1 is:",tuple_n)
#Concatenating a tuple t2=(11,13,15) with tuple_t1.
tuple_t2=(11,13,15)
tuple_t=tuple_t1+tuple_t2
print("Concatenated tuple_t i.e concatenation of tuple_t1 and tuple_t2 is:",tuple_t)
#returning maximum and minimum value from tuple_t.
a=max(tuple_t)
b=min(tuple_t)
print("maximum value from tuple_t is:",a)
print("minimum value from tuple_t is:",b)
```

OUTPUT

```
First half values of tuple t1 is: (1, 2, 5, 7, 9)
Other half values of tuple t1 is: (2, 4, 6, 8, 10)
Tuple whose values are even in tuple t1 is: (2, 2, 4, 6, 8, 10)
Concatenated tuple_t i.e concatenation of tuple_t1 and tuple_t2 is: (1, 2, 5, 7,
9, 2, 4, 6, 8, 10, 11, 13, 15)
maximum value from tuple_t is: 15
minimum value from tuple_t is: 1
```

Question 13: WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

CODE

```
def accept_name():
    try:
        # Get the name from the user
        name = input("Enter your name: ")

        # Check if the name contains digits or special characters
        if any(char.isdigit() or not char.isalnum() for char in name):
            raise ValueError("Name should not contain digits or special characters.")

        # If the name is valid, return it
        return name

    except ValueError as ve:
        # If an exception is raised, print the error message
        print(ve)
accept_name()
```

OUTPUT

```
=== RESTART: C:/Users/lenovo/AppData/Local/Programs/Python/Python311/py13.py ===
Enter your name: aman
Hello, aman
```