

9. Function

1. Code, execute and debug programs to solve the given problem using built in functions

```
print(abs(-7))
print(all({'*',' ',' '}))
print(all([' ',' ',' ']))
print(any((1,0,0)))
print(any((0,0,0)))
print(bin(7))
print(bool(0.5))
print(bool())
print(bytes(5))
a=bytearray([1,2,3,4,5])
print(a)
print(chr(65))
print(chr(97))
```

Output:

```
7
False
True
True
False
0b111
True
False
b'\x00\x00\x00\x00\x00'
bytearray(b'\x01\x02\x03\x04\x05')
A
a
```

2a. Code, execute and debug programs to solve the given problem by defining a function

```
def reverse(n):
    rev = 0
    while n > 0:
        r = n % 10
        rev = rev * 10 + r
        n = int(n / 10)
    return rev
x = int(input("Enter a number:"))
result = reverse(x)
print("Reverse is:", result)
```

Output:

```
Enter a number: 1234
Reverse is: 4321
```

2b. Python program to check if number is Palindrome

```
def Palindrome(n):
    reverse = 0
    reminder = 0
    while(n != 0):
        remainder = n % 10
        reverse = reverse * 10 + remainder
        n = int(n / 10)
    return reverse
```

```
num = int(input('Enter the number: '))
reverse = Palindrome(num)
if(num == reverse):
    print(num,'is a Palindrome')
else:
    print(num,'is not a Palindrome')
```

Output:

```
Enter the number: 545
545 is a Palindrome
Enter the number: 549
549 is not a Palindrome
```

2c. Python program to check if string is Palindrome

```
string = input('Enter the string: ')
i = string
reverse = ""
while(len(i) > 0):
    if(len(i) > 0):
        a = i[-1]
        i = i[:-1]
        reverse += a
if(reverse == string):
    print(string,'is a Palindrome')
else:
    print(string,'is not a Palindrome')
```

Output:

```
Enter the string: madam
madam is a Palindrome
Enter the string: polytechnic
polytechnic is not a Palindrome
```

3 a) Python program to calculate the factorial using recursion

```
def factorial(x):
    if x == 1:
        return 1
    else:
        return (x * factorial(x-1))
num = int(input("Enter the number"))
print("The factorial of", num, "is", factorial(num))
```

Output:

```
Enter the number 5
The factorial of 5 is 120
```

3b # Python program to find the sum of natural using recursive function

```
def recur_sum(n):
    if n <= 1:
        return n
    else:
        return n + recur_sum(n-1)
num = int(input("Enter the number"))
if num < 0:
    print("Enter a positive number")
```

```
else:  
    print("The sum is",recur_sum(num))
```

Output:

Enter the number 9
The sum is 45

3c. # Python program to display the Fibonacci sequence

```
def fibo(n):  
    if n <= 1:  
        return n  
    else:  
        return(fibo(n-1) + fibo(n-2))  
nterms = int(input("Enter the number"))  
if nterms <= 0:  
    print("Plese enter a positive integer")  
else:  
    print("Fibonacci sequence:")  
    for i in range(nterms):  
        print(fibo(i))
```

Output:

Enter the number 5
Fibonacci sequence:
0
1
1
2
3

4. Define anonymus function and code to solve the given problem**a) Demonstrate use of a lambda function to perform addition**

```
add=lambda num:num+4  
print(add(6))
```

Output:

10

b) Demonstrate use of a lambda function to reverse a string and convert to upper case

```
str1='GeeksforGeeks'  
rev_upper=lambda string:string.upper()[::-1]  
print(rev_upper(str1))
```

Output:

SKEEGROFSKEEG

c) Demonstrate use of a lambda function to find cube of the given number

```
lambda_cube=lambda y:y*y*y  
print("Using lambda function,cube:",lambda_cube(5))
```

Output:

Using lamda function,cube:125