Darshan UNIVERSITY

Python Programming - 2101CS405

Lab - 7

1

Functions

01) WAP to count simple interest using function.

```
def simpleIntrest(p, r, t):
    si = (p*r*t)/100
    return si

p = int(input("Enter Principle: "))
r = int(input("Enter Rate: "))
t = int(input("Enter Time: "))
# si = lambda p,r,t : (p*r*t)/100         #using lambda
print("====================================")
print("Simple Intrest is ", simpleIntrest(p,r,t))
# print("Simple Intrest is ", si(p,r,t))
```

```
Enter Principle: 120000
Enter Rate: 4
Enter Time: 2
====================================
Simple Intrest is  9600.0
```

### 02) WAP that defines a function to add first n numbers.

```
def addNum(n):
    i=1
    sum=0
    for i in range(n+1):
        sum = sum + i
    return sum

n = int(input("Enter number : "))
# sum = lambda n : sum(range(n+1))
# print("Sum of first",n ,"natuaral number is", sum(n))
print("====================================")
print("Sum of first",n ,"natuaral number is", addNum(n))
```

```
Enter number : 8
====================================
Sum of first 8 natuaral number is 36
```

03) WAP to find maximum number from given two numbers using function.

```
def maximum(a, b):
    if(a>b):
        return a
    else:
        return b

a = int(input("Enter first number : "))
b = int(input("Enter Second number : "))
print("====================================")
# ans = lambda a,b : max(a,b)
# print(ans(a,b))
print("Maximum Number is ", maximum(a,b))
```

```
Enter first number : 78
Enter Second number : 56
====================================
Maximum Number is  78
```

04) WAP that defines a function which returns 1 if the number is prime otherwise return 0.

```
def prime(n):
    if n == 0 or n == 1:
        return 0
    for i in range(2,n//2+1):
        if n%i==0:
            return 0
    return 1

print("'1' means to Prime \n'0' means to not Prime")
print("====================================")
n = int(input("Enter Number : "))
print("====================================")
print(prime(n))
```

```
'1' means to Prime
'0' means to not Prime
====================================
Enter Number : 897
====================================
0
```

05) Write a function called primes that takes an integer value as an argument and returns a list of all prime numbers up to that number.

```
def primes(n):
    for i in range(2,n+1):
        if prime(i)==1: # prime function is used from above code (4)
            li.append(i)
    return li

n = int(input("Enter Number : "))
li = []
print("====================================")
print("List of prime numbers upto ",n," is::\t", primes(n))
```

```
Enter Number : 9
====================================
List of prime numbers upto  9  is::      [2, 3, 5, 7]
```

06) WAP to generate Fibonacci series of N given number using function name fibbo. (e.g. 0 1 1 2 3 5 8...)

```
def fibonacci(n):
    for i in range(n+1):
        if n<=1:
            return n;
        return fibonacci(n-1)+fibonacci(n-2)

n = int(input("Enter Number : "))
print("====================================")
print("Fibonacci Series of",n,"elements is:: ")
```

```python
for i in range(n+1):
    print(fibonacci(i),end=" ")
```

```
Enter Number : 14
===================================
Fibonacci Series of 14 elements is::
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

07) WAP to find the factorial of a given number using recursion.

```python
def fact(n):
    if n<=1:
        return 1
    else:
        return fact(n-1)*n

n = int(input("Enter Number : "))
print("===================================")
# fact = lambda n: 1 if n<=1 else fact(n-1)*n   #using Lambda
print("Factorial of",n,"is",fact(n))
```

```
Enter Number : 7
===================================
Factorial of 7 is 5040
```

08) WAP to implement simple calculator using lamda function.

```python
n1 = int(input("Enter first number : "))
n2 = int(input("Enter second number : "))
print("===================================")
n3 = input("Choose Operation \t[+, -, *, /] : ")
ans = lambda n1,n2,n3 : n1+n2 if n3=="+" else n1-n2 if n3=="-" else n1*n2 if n3=="*" else n1//n2 if n3=="/" else "Invalid Operation"
print("===================================")
print("Answer is",ans(n1,n2,n3))
```

```
Enter first number : 56
Enter second number : 7
===================================
Choose Operation        [+, -, *, /] : *
===================================
Answer is 392
```

09)Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically

Sample Items : green-red-yellow-black-white

Expected Result : black-green-red-white-yellow

```python
st = input("Enter Hyphen Seperated String : ")
li = st.split("-")
li.sort()
print("===================================")
print("Output Sequence ::")
print("-".join(li))
```

```
Enter Hyphen Seperated String : green-red-yellow-black-white-Amber
===================================
Output Sequence ::
Amber-black-green-red-white-yellow
```

10) Write a python program to implement all function arguments type

Positional arguments

Default argument

Keyword arguments (named arguments)

Arbitrary arguments (variable-length arguments args and kwargs)

```python
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))
positionalArguments = lambda a,b : a*b
print("Product(By Positional arguments) : ",positionalArguments(a,b))
```

```
Enter First Number : 56
Enter Second Number : 10
```

```
Product(By Positional arguments) :  560
```

```python
a = int(input("Enter First Number : "))
defaultArgument = lambda a,b=10 : a*b
print("Product(By Default argument) : ",defaultArgument(a))
```

```
Enter First Number : 56
Product(By Default argument) :  560
```

```python
a = int(input("Enter First Number : "))
b = int(input("Enter Second Number : "))
def keywordArguments(a,b):
    return a*b
print("Product(By Keyword(or named) arguments) : ",keywordArguments(b=a,a=b))
```

```
Enter First Number : 56
Enter Second Number : 10
Product(By Keyword(or named) arguments) :  560
```

```python
a = int(input("Enter First Number : "))
def arbitraryArguments(a,*b):
    mul = a
    for i in b:
        mul*=i
    return mul
print("Product (By Arbitrary arguments) :",arbitraryArguments(a,10))
```

```
Enter First Number : 56
Product (By Arbitrary arguments) : 560
```

.11) WAP to calculate power of a number using recursion.

```python
def exp(base,power):
    if power==1:
        return base
    elif power==0:
        return 1
    else:
        return base*exp(base,power-1)
base = int(input("Enter Base : "))
power = int(input("Enter Power : "))
print("====================================")
print("Answer :: ",exp(base,power))
```

```
Enter Base : 8
Enter Power : 4
====================================
Answer ::  4096
```

12) WAP to count digits of a number using recursion.

```python
def countDigits(n):
    if n<10:
        return 1
    else:
        return 1+countDigits(n//10)

n = int(input("Enter Number : "))
ans = countDigits(n)
print("====================================")
print("Digit of",n,"is",ans)
```

```
Enter Number : 789456
====================================
Digit of 789456 is 6
```

13) WAP to reverse an integer number using recursion.

```python
rev_num = 0
def reverseNumber(n):
    global rev_num
    if(n>0):
        reminder = n%10
        rev_num = rev_num*10 + reminder
        reverseNumber(n//10)
    return rev_num

n = int(input("Enter Number : "))
rev_num = reverseNumber(n)
print("====================================")
print("Reverse : ",rev_num)
```

```
Enter Number : 4578
====================================
Reverse :  8754
```

14) WAP to convert decimal number into binary using recursion.

```
def decimalToBinary(n):
    if n==0:
        return 0
    else:
        return n%2+10*(decimalToBinary(n//2))

n = int(input("Enter Number : "))
ans = decimalToBinary(n)
print("====================================")
print("Binary form: ",ans)
```

```
Enter Number : 54
====================================
Binary form:  110110
```