

Recursion Programs

1. Write a Python Program to determine whether a given number is even or odd recursively.

Answer:

```
def check(n):
    if (n < 2):
        return (n % 2 == 0)
    return (check(n - 2))

n=int(input("Enter number:"))
if(check(n)==True):
    print("Number is even!")
else:
    print("Number is odd!")
```

2. Write a Python Program to determine how many times a given letter occurs in a string recursively.

Answer:

```
def check(string,ch):
    if not string:
        return 0
    elif string[0]==ch:
        return 1+check(string[1:],ch)
    else:
        return check(string[1:],ch)

string=input("Enter string:")
ch=input("Enter character to check:")
print("Count is:")
print(check(string,ch))
```

3. Write a Python Program to find the fibonacci series using recursion.

Answer:

```
def fibonacci(n):
    if(n <= 1):
        return n
    else:
        return(fibonacci(n-1) + fibonacci(n-2))
n = int(input("Enter number of terms:"))
```

```
print("Fibonacci sequence:")
for i in range(n):
    print(fibonacci(i), end=" ")
```

4. Write a Python Program to find the factorial of a number using recursion.

Answer:

```
def factorial(n):
    if(n <= 1):
        return 1
    else:
        return(n*factorial(n-1))
n = int(input("Enter number:"))
print("Factorial:")
print(factorial(n))
```

5. Write a Python Program to find the sum of elements in a list recursively.

Answer:

```
def sum_arr(arr,size):
    if (size == 0):
        return 0
    else:
        return arr[size-1] + sum_arr(arr,size-1)
n=int(input("Enter the number of elements for list:"))
a=[]
for i in range(0,n):
    element=int(input("Enter element:"))
    a.append(element)
print("The list is:")
print(a)
print("Sum of items in list:")
b=sum_arr(a,n)
print(b)
```

6. Write a Python Program to find the binary equivalent of a number recursively.

Answer:

```
l=[]
def convert(b):
    if(b==0):
```

```

    return l
    dig=b%2
    l.append(dig)
    convert(b//2)
a=int(input("Enter a number: "))
convert(a)
l.reverse()
print("Binary equivalent:")
for i in l:
    print(i)

```

7. Write a Python Program to find the sum of the digits of the number recursively.

Answer:

```

l=[]
def sum_digits(b):
    if(b==0):
        return l
    dig=b%10
    l.append(dig)
    sum_digits(b//10)
n=int(input("Enter a number: "))
sum_digits(n)
print(sum(l))

```

8. Write a Python Program to find the LCM of two numbers using recursion.

Answer:

```

def lcm(a,b):
    lcm.multiple=lcm.multiple+b
    if((lcm.multiple % a == 0) and (lcm.multiple % b == 0)):
        return lcm.multiple;
    else:
        lcm(a, b)
    return lcm.multiple
lcm.multiple=0
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
if(a>b):
    LCM=lcm(b,a)
else:
    LCM=lcm(a,b)

```

```
print(LCM)
```

9. Write a Python Program to find the GCD of two numbers using recursion.

Answer:

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
print("GCD is: ")
print(GCD)
```

10. Write a Python Program to find if a number is prime or not using recursion.

Answer:

```
def check(n, div = None):
    if div is None:
        div = n - 1
    while div >= 2:
        if n % div == 0:
            print("Number not prime")
            return False
        else:
            return check(n, div-1)
    else:
        print("Number is prime")
        return True
n=int(input("Enter number: "))
check(n)
```

11. Write a Python Program to find the product of two numbers using recursion.

Answer:

```
def product(a,b):
    if(a<b):
        return product(b,a)
    elif(b!=0):
```

```
        return(a+product(a,b-1))
    else:
        return 0
a=int(input("Enter first number: "))
b=int(input("Enter second number: "))
print("Product is: ",product(a,b))
```

12. Write a Python Program to find the power of a number using recursion.

Answer:

```
def power(base,exp):
    if(exp==1):
        return(base)
    if(exp!=1):
        return(base*power(base,exp-1))
base=int(input("Enter base: "))
exp=int(input("Enter exponential value: "))
print("Result:",power(base,exp))
```

13. Write a Python Program to check whether a string is a palindrome or not using recursion.

Answer:

```
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Enter string:"))
if(is_palindrome(a)==True):
    print("String is a palindrome!")
else:
    print("String isn't a palindrome!")
```

14. Write a Python Program to reverse a string using recursion.

Answer:

```
def reverse(string):
    if len(string) == 0:
```

```

    return string
else:
    return reverse(string[1:]) + string[0]
a = str(input("Enter the string to be reversed: "))
print(reverse(a))

```

15. Write a Python Program to flatten a nested list using recursion.

Answer:

The **isinstance()** function returns True if the specified object is of the specified type, otherwise False . If the # type parameter is a list, this function will return True if the object is one of the types in the list.

```

def flatten(S):
    if S == []:
        return S
    if isinstance(S[0], list):
        return flatten(S[0]) + flatten(S[1:])
    return S[:1] + flatten(S[1:])
s=[[1,2],[3,4]]
print("Flattened list is: ",flatten(s))

```

16. Write a Python Program to find the total sum of a nested list using recursion.

Answer:

```

def sum1(lst):
    total = 0
    for element in lst:
        if (type(element) == type([])):
            total = total + sum1(element)
        else:
            total = total + element
    return total
print( "Sum is:",sum1([[1,2],[3,4]]))

```

17. Write a Python Program to find the length of a list recursively.

Answer:

```

def length(lst):
    if not lst:
        return 0
    return 1 + length(lst[1::2]) + length(lst[2::2])

```

```
a=[1,2,3]
print("Length of the string is: ")
print(a)
```