NAME:ROHAN NYATI

SAP ID:500075940

ROLL NO. : R177219148

BATCH-5(AI&ML)

Algorithm for Intelligent System and Robotic Lab

Practical 5

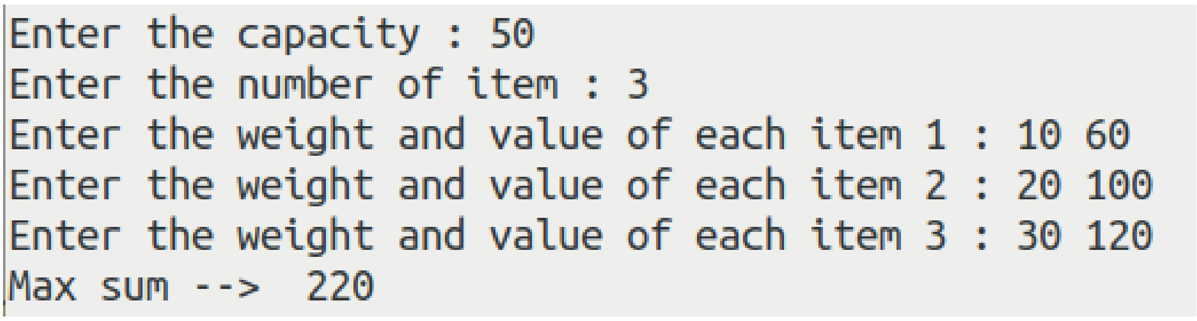
Aim – To implement a solution for any NP Complete problem using recursion.

Problem Chosen : 0-1Knapsack Problem

--- Implementation in Python3 ---

```
class item :
    def __init__( self , weight , value ) :
            self.value = value
            self.weight = weight

def knapsack( items , capacity , n ) :
    if( n==0 ) :
        return 0 ;
    elif( capacity < items[n-1].weight ):
        return knapsack( items , capacity , n - 1 ) ;
    else :
        sack_with_curr_item = knapsack( items , capacity -
items[n - 1].weight , n - 1 ) + items[ n - 1 ].value
        sack_without_curr_item = knapsack( items , capacity ,
n - 1 )
        return max( sack_with_curr_item ,
sack_without_curr_item )


if __name__ == "__main__" :
    items = []
    capacity = int(input("Enter the capacity : "))
    n =  int(input("Enter the number of item : "))


    for i in range( n ) :
        items.append( item( *( map(int , input(f"Enter the
weight and value of each item {i+1} : ").split() ) ) ) )


    print("Max sum --> " , knapsack( items , capacity , n ))
```

--- OUTPUT ---



```
Enter the capacity : 50
Enter the number of item : 3
Enter the weight and value of each item 1 : 10 60
Enter the weight and value of each item 2 : 20 100
Enter the weight and value of each item 3 : 30 120
Max sum -->  220
```