Genetic Algorithm & Fuzzy Logic

Semester-5

Practical-1

Name: Shivam Tawari

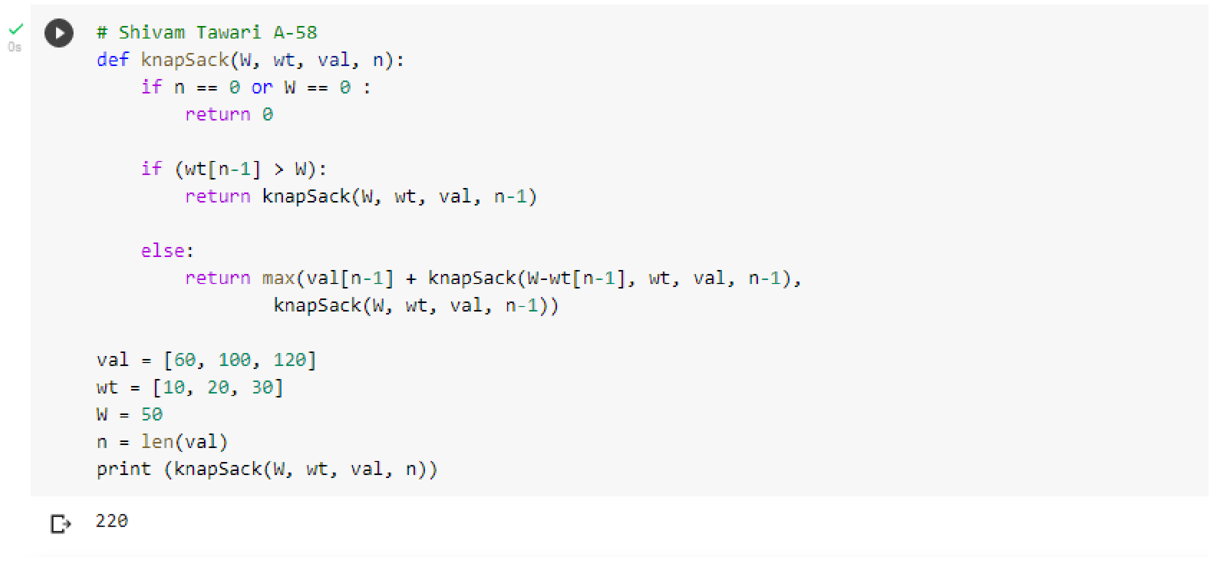
Roll no: A-58

Aim: Implement binary encoding for solving Knapsack problem.

Theory:

The knapsack problem is popular in the research field of constrained and combinatorial optimization with the aim of selecting items into the knapsack to attain maximum profit while simultaneously not exceeding the knapsack's capacity. Genetic algorithms (GAs) are stochastic search algorithms that mimic the biological process of evolution enabling thereby users to solve complex optimization problems. They operate based on a population of chromosomes, where a chromosome represents a candidate solution. Genetic operators allow the population to evolve over time and to converge to the optimal solution. In the spirit of "survival of the fittest," GAs is naturally designed to solve maximization problems in which chromosomes with high fitness are more likely to survive and the objective function is termed fitness function. The best-known GA design, where chromosomes are represented as a binary string like '1010001'. The zeros and ones are the genes of a binary chromosome, and their meaning depends on the problem one tries to solve. Binary encoding is the most common one, mainly because the first research of GA used this type of encoding and because of its relative simplicity. Encoding of chromosomes is the first question to ask when starting to solve a problem with GA. Encoding depends on the problem heavily.

Code:



```python
# Shivam Tawari A-58
def knapSack(W, wt, val, n):
    if n == 0 or W == 0 :
        return 0

    if (wt[n-1] > W):
        return knapSack(W, wt, val, n-1)

    else:
        return max(val[n-1] + knapSack(W-wt[n-1], wt, val, n-1),
                   knapSack(W, wt, val, n-1))

val = [60, 100, 120]
wt = [10, 20, 30]
W = 50
n = len(val)
print (knapSack(W, wt, val, n))
```

```
220
```

Conclusion: Hence Knapsack algorithm has been implemented successfully.