



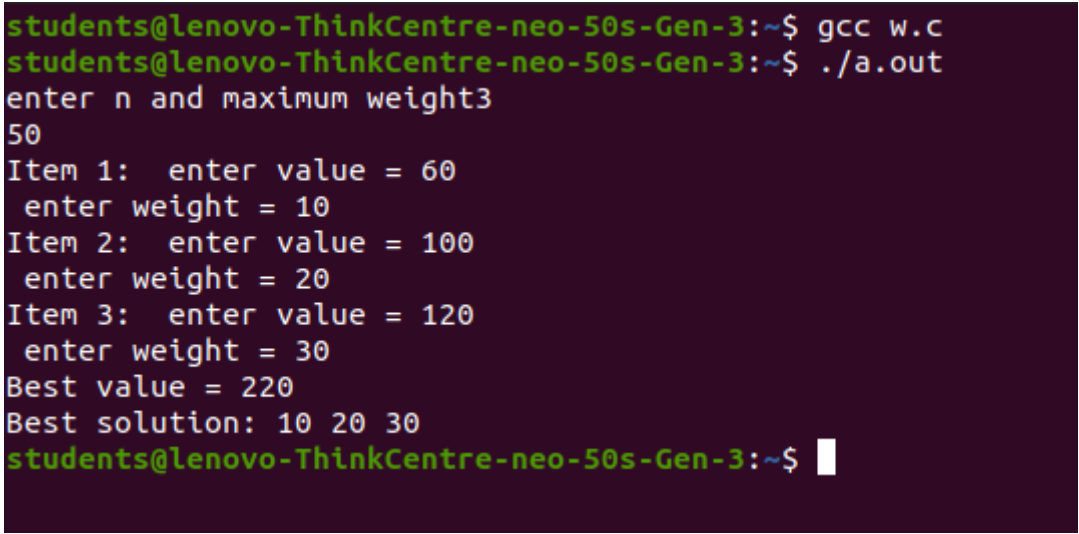
Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

PRACTICAL NO. 08	DESIGN AND ANALYSIS OF ALGORITHMS
NAME	VEDANTI ANIL WADATKAR
UID	2021700072
BATCH	D4
PROBLEM STATEMENT	Branch and bound (To implement 0/1 Knapsack problem using Branch and Bound.)
CODE	<pre>#include <stdio.h> #include <stdlib.h> #define MAX_ITEMS 100 #define MAX_WEIGHT 1000 struct item { int value; int weight; }; struct node { int level; int profit; int weight; }; int n; int max_weight; int best_value; int solution[MAX_ITEMS]; struct item items[MAX_ITEMS];</pre>

```
void knapsack() {
    struct node stack[MAX_ITEMS * 2];
    int top = 0;
    struct node current = {0, 0, 0};
    stack[top++] = current;
    while (top > 0) {
        current = stack[--top];
        if (current.level == n) {
            if (current.profit > best_value) {
                best_value = current.profit;
                for (int i = 0; i < n; i++) {
                    solution[i] = items[i].weight;
                }
            }
            continue;
        }
        if (current.weight + items[current.level].weight <= max_weight) {
            struct node child = {current.level + 1,
                                current.profit + items[current.level].value,
                                current.weight + items[current.level].weight};
            stack[top++] = child;
        }
        struct node child = {current.level + 1,
                            current.profit,
                            current.weight};
        stack[top++] = child;
    }
}

int main() {
    printf("enter n and maximum weight");
```

	<pre> scanf("%d%d", &n, &max_weight); for (int i = 0; i < n; i++) { printf("Item %d: ", i + 1); printf(" enter value = "); scanf("%d", &items[i].value); printf(" enter weight = "); scanf("%d", &items[i].weight); } knapsack(); printf("Best value = %d\n", best_value); printf("Best solution:"); for (int i = 0; i < n; i++) { printf(" %d", solution[i]); } printf("\n"); return 0; } </pre>
OUTPUT	 <pre> students@lenovo-ThinkCentre-neo-50s-Gen-3:~\$ gcc w.c students@lenovo-ThinkCentre-neo-50s-Gen-3:~\$./a.out enter n and maximum weight3 50 Item 1: enter value = 60 enter weight = 10 Item 2: enter value = 100 enter weight = 20 Item 3: enter value = 120 enter weight = 30 Best value = 220 Best solution: 10 20 30 students@lenovo-ThinkCentre-neo-50s-Gen-3:~\$ █ </pre>

