Report No: 01

Report Name: WAP to find out maximum profit and show knapsack Array of taken products by Greedy Design Algorithm

Code:

```
#include<iostream>
using namespace std;

void knapsack(int n, float weight[], float profit[], float capacity) {
  float x[n], tp = 0;
  int i, j, u;
  u = capacity;

  for (i = 0; i < n; i++)
    x[i] = 0.0;

  for (i = 0; i < n; i++) {
    if (weight[i] > u)
      break;
    else {
      x[i] = 1.0;
      tp = tp + profit[i];
      u = u - weight[i];
    }
  }

  if (i < n) x[i] = u / weight[i];

  tp = tp + (x[i] * profit[i]);

  cout << "Unit per taken: " << endl;
  for (i = 0; i < n; i++){
    cout << "Object no " << (i+1) << " is: " << (x[i] * weight[i]) << " kg" <<endl;
  }
  cout << "\nMax profit: " << tp ;

}

int main() {
  int num, i, j;
  cout << "Total No. of objects: ";
  cin >> num;
  float weight[num], profit[num], capacity;
  float ratio[num], temp;

  cout << "Enter Weight and profits for each object: " <<endl;
  for (i = 0; i < num; i++) {
    cout << (i+1) << " No. object Profit is: ";
    cin >> profit[i];
    cout << (i+1) << " No. object Weight is: ";
    cin >> weight[i];
  }

  cout << "Enter total capacity: ";
  cin >> capacity;

  for (i = 0; i < num; i++) {
    ratio[i] = profit[i] / weight[i];
  }

  for (i = 0; i < num; i++) {
    for (j = i + 1; j < num; j++) {
      if (ratio[i] < ratio[j]) {
        temp = ratio[j];
        ratio[j] = ratio[i];
        ratio[i] = temp;

        temp = weight[j];
        weight[j] = weight[i];
        weight[i] = temp;

        temp = profit[j];
        profit[j] = profit[i];
        profit[i] = temp;
      }
    }
  }

  knapsack(num, weight, profit, capacity);

}
```

Input :

```
Total No. of objects: 5
Enter Weight and profits for each object:
1 No. object Profit is: 10
1 No. object Weight is: 3
2 No. object Profit is: 15
2 No. object Weight is: 3
3 No. object Profit is: 10
3 No. object Weight is: 2
4 No. object Profit is: 12
4 No. object Weight is: 5
5 No. object Profit is: 8
5 No. object Weight is: 1
Enter total capacity: 5
```

Output:

```
Unit per taken:
Object no 1 is: 1 kg
Object no 2 is: 2 kg
Object no 3 is: 2 kg
Object no 4 is: 0 kg
Object no 5 is: 0 kg

Max profit: 28
```