|  |  |
| --- | --- |
| **Experiment No.** | **5** |
| **Aim** | **Experiment using greedy approach (Knapsack problem)** |
| **Name** | **Rucha Sudhir Kulkarni** |
| **UID No.** | **2021300067** |
| **Class & Division** | **SE Computer Engineering (Div:A)(Batch:D)** |
| **Date of Performance:** | **29.03.2023** |
| **Date of Submission:** | **04.04.2023** |

**Aim: – To implement solution to Knapsack problem(fractional) using greedy approach.**

**Algorithm:**

1. Start
2. Input number of items n, their capacities and values.
3. Input the limit also.
4. Print the values in table format.
5. Compute ratio[i]=capacity[i]/value[i].
6. Arrange all the items in descending order of the capacity[i]/value[i] ratio.
7. Print the new table.
8. Cap=0, profit=0.0
9. From this new table pick items one by one.
   1. If cap+capacity[i]<=limit
   2. Add the item to ans array
   3. Cap=cap+capacity[i]
   4. Profit=profit+value[i]
   5. Else
   6. Profit=profit+(limit-cap)\*value[i]/(capacity[i])
   7. Add the item to ans array
   8. Break the loop.
10. Print the profit.
11. Print the elements of ans array.
12. stop

**Program:**

#include<stdio.h>

int capacity[100],value[100];

int n;

int main(){

printf("Enter number of items");

scanf("%d",&n);

printf("Enter capacities/weights\n ");

for(int i=0;i<n;i++){

scanf("%d",&capacity[i]);

}

printf("Enter values\n ");

for(int i=0;i<n;i++){

scanf("%d",&value[i]);

}

printf("\n");

printf("Capacity array={ ");

for(int i=0;i<n;i++){

printf("%d, ",capacity[i]);

}

printf("}\n");

printf("Value array={ ");

for(int i=0;i<n;i++){

printf("%d, ",value[i]);

}

int itemID[n];

for(int i=0;i<n;i++){

    itemID[i]=i;

}

printf("}\n");

int limit;

printf("Enter limit");

scanf("%d",&limit);

float ratio[n];

for(int i=0;i<n;i++){

    ratio[i]=(float)value[i]/(float)capacity[i];

}

printf("Initial table:\n");

printf("ItemId Weight Value  Ratio\n");

for(int i=0;i<n;i++){

    printf("%d\t%d\t%d\t%f\n",itemID[i],capacity[i],value[i],ratio[i]);

}

for(int i=0;i<n;i++){

    for(int j=i+1;j<n;j++){

        if(ratio[i]<ratio[j]){

            float temp=ratio[i];

            ratio[i]=ratio[j];

            ratio[j]=temp;

            int t=capacity[i];

            capacity[i]=capacity[j];

            capacity[j]=t;

            t=value[i];

            value[i]=value[j];

            value[j]=t;

            t=itemID[i];

            itemID[i]=itemID[j];

            itemID[j]=t;

        }

    }

}

printf("New table:\n");

printf("ItemId Weight Value  Ratio\n");

for(int i=0;i<n;i++){

    printf("%d\t%d\t%d\t%f\n",itemID[i],capacity[i],value[i],ratio[i]);

}

int ans[n],ele=0;

int cap=0,j=0;

float profit=0.0;

for(int i=0;i<n;i++){

    if(cap+capacity[i]<=limit){

        cap+=capacity[i];

        ans[j]=itemID[i];

        profit+=value[i];

        j++;

    }

    else{

        profit+=((limit-cap)\*value[i])/(float)(capacity[i]);

        ans[j]=itemID[i];

        ele=j;

        break;

    }

}

printf("Total profit is:%f\n",profit);

printf("Elements that are included are: ");

for(int i=0;i<=j;i++){

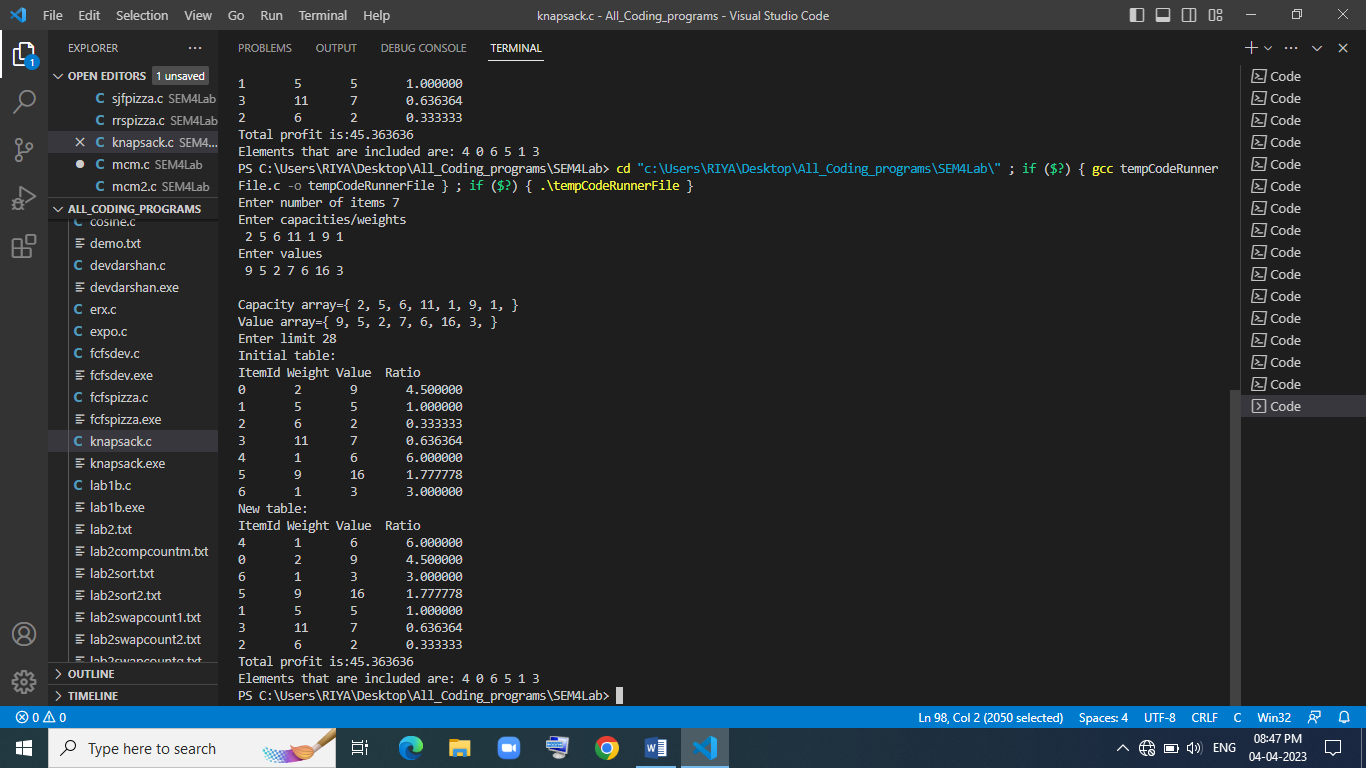
    printf("%d ",ans[i]);

}

return 0;

}

**Output and Observation:**



Knapsack problem solution has been successfully implemented in C language and its solution has been verified manually on notebook.

**Conclusion:**

After performing the above experiment, I got to know how to what is knapsack problem and how to find its solution using greedy approach.