

Experiment - 5

W.A.P in c language to implement Fractional knapsack

```
#include <stdio.h>
#include <conio.h>
void knapsack (int n, float weight [], float profit [], float capacity, float x []) {
    float u;
    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;
            u = u - weight[i];
        }
    }
    if (i < n) {
        x[i] = u / weight[i];
    }
}
```

Teacher's Signature _____


```
    tp = tp + (xt[i] * profit[i]);  
    for (i=0; i<n; i++)  
        printf("%f\t", x[i]);  
    printf("\n Maximum profit is = %f", tp); }
```

```
int main() {  
    float weight[20], profit[20], capacity, ratio[20], temp;  
    int num;  
    printf("\n Enter the no. of objects :-");  
    scanf("%d", &num);  
    printf("\n Enter the weight and profits of each object :-");  
    for (int i=0; i<num; i++)  
        scanf("%f %f", &weight[i], &profit[i]);  
    printf("\n Enter the Capacity of knapsack =");  
    scanf("%f", &capacity);
```

```
    for (int i=0; i<num; i++)  
    {  
        ratio[i] = profit[i] / weight[i];  
    }  
    for (i=0; i<num; i++)  
    {  
        for (int j=i+1; j<num; j++)  
        {  
            if (ratio[i] < ratio[j])
```


Expt. No.

if

```
temp = ratio[j];  
ratio[j] = ratio[i];  
ratio[i] = temp;
```

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

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

```
}  
}  
}
```

```
knapack (num, weight, profit, capacity);  
}
```

Teacher's Signature

Output.

Enter the no. of objects = 7

Enter the weight and profits of each objects:- 2 10

3 5

5 15

7 7

1 6

4 18

1 3

Enter the Capacity of knapsack: 15

1.000000 1.000000 1.000000 1.000000 1.000000
0.666667 0.000000

Maximum profit is : 55.3332.