EXPERIMENT - 06

PROGRAM:-

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define MIN(a, b) (a < b ? a : b)
typedef struct job_ {
int id, deadline, profit;
}job;
int compare(const void *a, const void *b) { // descending
return (((job *) a)->profit < ((job *) b)->profit);
}
void schedule(job data[], int n) {
int i, j, check[n], ans = 0;
memset(check, 0, sizeof(check));
for (i = 0; i < n; i++) {
for (j = MIN(data[i].deadline, n)-1; j >= 0; j--) {
if (!check[j]) {
check[j] = data[i].id;
ans += data[i].profit;
break;
}
}
}
printf("The sequence of job is: \n");
for (i = 0; i < n; i++)
if (check[i])
printf("%d ", check[i]);
printf("\nThe max profit is: %d\n", ans);
}
int main() {
job data[10];
int n, i, j;
printf("Enter number of jobs:\n");
scanf("%d", &n);
printf("Enter jobs in the order (id deadline profit):\n");
for (i = 0; i < n; i++)
scanf("%d%d%d", &data[i].id, &data[i].deadline, &data[i].profit);
qsort(data, n, sizeof(job), compare);
schedule(data, n);
return 0;
}
```

OUTPUT:-

```
(base) computer@computer:~/Desktop/satyaprakash_pal$ gedit exp6.c
(base) computer@computer:~/Desktop/satyaprakash_pal$ gcc exp6.c
(base) computer@computer:~/Desktop/satyaprakash_pal$ ./a.out
Enter number of jobs:
5
Enter jobs in the order (id deadline profit):
a 2 100
The sequence of job is:
-1058402120
The max profit is: 15775487
(base) computer@computer:~/Desktop/satyaprakash_pal$ ./a.out
Enter number of jobs:
5
Enter jobs in the order (id deadline profit):
1 2 100 2 1 19 3 2 27 4 1 25 5 3 15
The sequence of job is:
3 1 5
The max profit is: 142
```

EXPERIMENT - 05

PROGRAM:-

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
// Structure to store the values of the knapsack object
typedef struct
{
char name[20];
int profit;
int weight;
float ratio;
}knapsack;
// Sort object in descending order by the profit/weight ratio
knapsack *sort(knapsack *arr, int start, int end)
{
int size = (end - start);
if(size == 1) return arr+start;
int mid = (start + end) / 2;
knapsack *lft = sort(arr, start, mid);
knapsack *rht = sort(arr, mid, end);
int l_size = (mid - start);

int r_size = (end - mid);
knapsack *left = malloc(sizeof(knapsack) * l_size);
knapsack *right = malloc(sizeof(knapsack) * r_size);
for(int i=0;i<l_size;i++)
{
left[i] = lft[i];
}
for(int i=0;i<r_size;i++)
{
right[i] = rht[i];
}
int l=0, r=0, a=start;
while((l < l_size) && (r < r_size))
{
if(left[l].ratio > right[r].ratio)
{
arr[a] = left[l];
l++;
}
else
{
arr[a] = right[r];
r++;
}
a++;
}
if(l == l_size)
```

```
{
while(r < r_size)
{
arr[a] = right[r];
r++;
a++;
}
}
else
{
while(l < l_size)
{
arr[a] = left[l];
l++;
a++;
}
}
return arr+start;
}

int main()
{
int total_objects, capacity;
printf("Enter Total Number of Objects: ");
scanf("%d", &total_objects);
printf("Enter Capacity of Knapsack: ");
scanf("%d", &capacity);
knapsack *obj = malloc(sizeof(knapsack) * total_objects);
// Gets the info of the objects
for(int i=0; i<total_objects; i++)
{
printf("\nEnter Object Name: ");
scanf("%s", obj[i].name);
getchar();
printf("Enter %s Profit: ", obj[i].name);
scanf("%d", &obj[i].profit);
printf("Enter %s Weight: ", obj[i].name);
scanf("%d", &obj[i].weight);
obj[i].ratio = (float) obj[i].profit / obj[i].weight;
}
sort(obj, 0, total_objects); // Sort the object
float profit = 0.0; // Stores the total profit
int div;
for(int i=0;i<total_objects;i++)
{
if(capacity <= 0) break;
if(capacity > obj[i].weight)
{
div = obj[i].weight;
}
else
{
```

```
div = capacity;
}
profit += (((float)div / obj[i].weight) * obj[i].profit);// Calculate the profit
capacity -= div;
}
printf("Total Profit: %.3f\n", profit); // Print the total profit
return 0;
}
```

OUTPUT:-

```
(base) computer@computer:~/Desktop/satyaprakash_pal$ ./a.out
Enter Total Number of Objects: 6
Enter Capacity of Knapsack: 5

Enter Object Name: A
Enter A Profit: 56
Enter A Weight: 22

Enter Object Name: B
Enter B Profit: 44
Enter B Weight: 23

Enter Object Name: C
Enter C Profit: 12
Enter C Weight: 1

Enter Object Name: D
Enter D Profit: 55
Enter D Weight: 20

Enter Object Name: E
Enter E Profit: 69
Enter E Weight: 21

Enter Object Name: F
Enter F Profit: 14
Enter F Weight: 5
Total Profit: 25.143
(base) computer@computer:~/Desktop/satyaprakash_pal$
```