Practical -6

## Aim:-

# Implement Program for fractional knapsack using Greedy design technique.

## CODE:

## #include <stdio.h>

## #include <stdlib.h>

## struct Item

## {

## int value;

## int weight;

## float density;

## };

## int compare(const void\* a, const void\* b)

## {

## struct Item\* item1 = (struct Item\*)a;

## struct Item\* item2 = (struct Item\*)b;

## return item2->density - item1->density;

## }

## float knapsack(int capacity, struct Item items[], int n)

## {

## qsort(items, n, sizeof(struct Item), compare);

## 

## float max\_value = 0.0;

## 

## 

## for (int i = 0; i < n && capacity > 0; i++)

## {

## if (items[i].weight <= capacity)

## {

## max\_value += items[i].value;

## capacity -= items[i].weight;

## }

## else

## {

## max\_value += (capacity \* items[i].density);

## capacity = 0;

## }

## }

## 

## return max\_value;

## }

## int main()

## {

## int capacity = 50;

## struct Item items[] = {{60, 10}, {100, 20}, {120, 30}};

## int n = sizeof(items) / sizeof(items[0]);

## 

## float max\_value = knapsack(capacity, items, n);

## printf("Maximum value that can be obtained = %f", max\_value);

## 

## return 0;

## }

## OUTPUT:

