**EXP 1**  
#include <stdio.h>

int main() {

int arr[100];

int n, i, j, minIndex, temp;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

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

}

for (i = 0; i < n - 1; i++) {

minIndex = i;

for (j = i + 1; j < n; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

}

printf("Sorted array: ");

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

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

}

printf("\n");

return 0;

}

**EXP2  
#include <stdio.h>**

**void insertionSort(int arr[], int n) {**

**for (int i = 1; i < n; i++) {**

**int key = arr[i], j = i - 1;**

**while (j >= 0 && arr[j] > key) {**

**arr[j + 1] = arr[j];**

**j--;**

**}**

**arr[j + 1] = key;**

**}**

**}**

**int main() {**

**int n;**

**printf("Enter number of elements: ");**

**scanf("%d", &n);**

**int arr[n];**

**printf("Enter %d elements:\n", n);**

**for (int i = 0; i < n; i++) scanf("%d", &arr[i]);**

**insertionSort(arr, n);**

**printf("Sorted array: ");**

**for (int i = 0; i < n; i++) printf("%d ", arr[i]);**

**return 0;**

**}**

**Exp3**

**#include<stdio.h>**

**int binarysearch( int arr[] ,int size,int key);**

**int main()**

**{**

**int i,size,key;**

**int arr[100];**

**printf("Enter the number of elements in array:");**

**scanf("%d",&size);**

**printf("Enter the elements in sorted manner:");**

**for(i=1;i<size+1;i++)**

**{**

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

**}**

**printf("Enter the element that you want to search:");**

**scanf("%d",&key);**

**int result = binarysearch(arr,size,key);**

**if (result!=-1)**

**printf("We found %d element in %d position",key,result);**

**else**

**printf("element no found");**

**}**

**int binarysearch( int arr[] ,int size,int key)**

**{**

**int low = 1, high = size;**

**while (low <= high) {**

**int mid = (low + high) / 2;**

**if (arr[mid] == key) {**

**return mid;**

**} else if (arr[mid] < key) {**

**low = mid + 1;**

**} else {**

**high = mid - 1;**

**}**

**}**

**return -1;**

**}**

**EXP4  
#include <stdio.h>**

**void merge(int arr[], int l, int m, int r) {**

**int n1 = m - l + 1;**

**int n2 = r - m;**

**int L[n1], R[n2];**

**for (int i = 0; i < n1; i++) L[i] = arr[l + i];**

**for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];**

**int i = 0, j = 0, k = l;**

**while (i < n1 && j < n2) {**

**if (L[i] <= R[j]) arr[k++] = L[i++];**

**else arr[k++] = R[j++];**

**}**

**while (i < n1) arr[k++] = L[i++];**

**while (j < n2) arr[k++] = R[j++];**

**}**

**void mergeSort(int arr[], int l, int r) {**

**if (l < r) {**

**int m = l + (r - l) / 2;**

**mergeSort(arr, l, m);**

**mergeSort(arr, m + 1, r);**

**merge(arr, l, m, r);**

**}**

**}**

**int main() {**

**int n;**

**printf("Enter number of elements: ");**

**scanf("%d", &n);**

**int arr[n];**

**printf("Enter %d elements:\n", n);**

**for (int i = 0; i < n; i++) scanf("%d", &arr[i]);**

**mergeSort(arr, 0, n - 1);**

**printf("Sorted array: ");**

**for (int i = 0; i < n; i++) printf("%d ", arr[i]);**

**return 0;**

**}**

**EXP5**

**#include<stdio.h>**

**#include<stdlib.h>**

**struct Item{**

**int weight;**

**int value;**

**float ratio;**

**};**

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

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

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

**if(item1->ratio<item2->ratio)**

**return 1;**

**else**

**return -1;**

**}**

**float fractionalKnapsack(int capacity,struct Item items[],int n) {**

**qsort(items,n,sizeof(items[0]),compare);**

**int currentWeight=0;**

**float totalValue=0.0;**

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

**if(currentWeight +items[i].weight<=capacity){**

**currentWeight +=items[i].weight;**

**totalValue+=items[i].value;**

**}else{**

**int remainingWeight=capacity-currentWeight;**

**totalValue+=items[i].value\*((float)remainingWeight/items[i].weight);**

**break;**

**}**

**}**

**return totalValue;**

**}**

**int main(){**

**int n,capacity;**

**printf("Enter number of items:");**

**scanf("%d",&n);**

**printf("Enter the capacity of Knapsack:");**

**scanf("%d",&capacity);**

**struct Item items[n];**

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

**printf("Enter value and weight for item %d:",i+1);**

**scanf("%d %d",&items[i].value,&items[i].weight);**

**items[i].ratio=(float)items[i].value/items[i].weight;**

**}**

**float maxValue=fractionalKnapsack(capacity,items,n);**

**printf("Maximum value in Knapsack=%.2f\n",maxValue);**

**return 0;**

**}**