NAME- Aniket Kalbhor

PRN- 12210601

ROLL- CSB-48

CODE- **2.A Write a program to implement Heap sort on structure that have a member a.Rollno and b.Name.**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct student

{

int rno;

char names[20];

};

void accept\_details(struct student arr[], int n)

{

printf("Enter the roll number followed by name\n");

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

{

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

scanf("%s", arr[i].names);

}

}

void print\_details(struct student arr[], int n)

{

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

{

printf("%d\n", arr[i].rno);

printf("%s\n", arr[i].names);

}

}

void create\_maxheap\_rno(struct student arr[], int i, int n)

{

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left].rno > arr[largest].rno)

largest = left;

if (right < n && arr[right].rno > arr[largest].rno)

largest = right;

if (i != largest)

{

struct student temp = arr[i];

arr[i] = arr[largest];

arr[largest] = temp;

create\_maxheap\_rno(arr, largest, n);

}

}

void heapsort\_rno(struct student arr[], int n)

{

for (int i = n / 2 - 1; i >= 0; i--)

create\_maxheap\_rno(arr, i, n);

for (int i = n - 1; i > 0; i--)

{

struct student temp = arr[0];

arr[0] = arr[i];

arr[i] = temp;

create\_maxheap\_rno(arr, 0, i);

}

}

int main()

{

struct student arr[10];

int n = 5;

accept\_details(arr, n);

printf("Accepted details:\n\n");

print\_details(arr, n);

printf("\n\n");

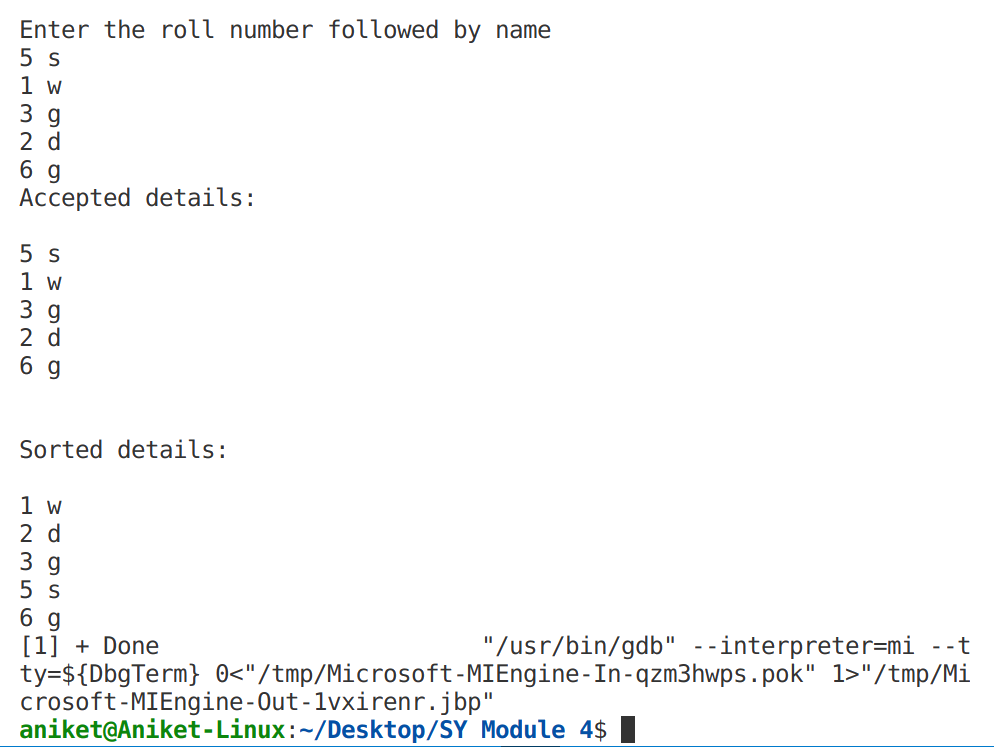
heapsort\_rno(arr, n);

printf("Sorted details:\n\n");

print\_details(arr, n);

return 0;

}

OUTPUT-

CODE- **2.B  Write a program to implement Quick sort on structure that have a member a.Rollno and b.Name.**

**#include <stdio.h>**

struct student

{

int rno;

char names[20];

};

void accept\_details(struct student arr[], int n)

{

printf("Enter the roll number followed by name\n");

for (int i = 0; i < n+1; i++)

{

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

scanf("%s", arr[i].names);

}

}

void print\_details(struct student arr[], int n)

{

for (int i = 0; i < n+1; i++)

{

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

printf("%s\n", arr[i].names);

}

}

void swap(struct student x, struct student y){

struct student temp = x;

x = y;

y = temp;

}

int partition(struct student arr[], int low, int high){

int pivot = low;

int i = low+1;

int j = high;

while (i<high)

{

while (i<=j)

if(arr[i].rno<arr[pivot].rno)

i++;

while (i<=j)

if(arr[j].rno>arr[pivot].rno)

j--;

if(i<j)

swap(arr[i], arr[j]);

}

swap(arr[j], arr[low]);

return j;

}

void quick\_sort(struct student arr[], int low, int high){

if (low != high)

{

int x = partition(arr, low, high);

quick\_sort(arr, low, x-1);

quick\_sort(arr, x+1, high);

}

}

int main(){

struct student arr[5];

int n = 4;

accept\_details(arr, n);

printf("\nAccepted details:\n");

print\_details(arr, n);

printf("\n\n");

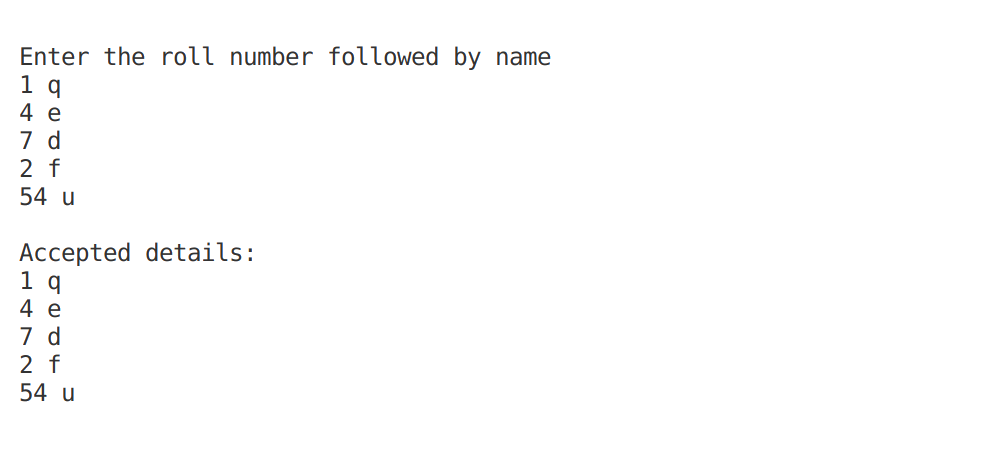
quick\_sort(arr, 0, n);

printf("\nSorted details:\n");

print\_details(arr, n);

return 0;

}

OUTPUT-