**Assignment 2**

**Name** : Komal Mahadev Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

1. **Write a program to implement Heap sort on structure that have a member a.Rollno and b.Name.**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  struct Student {      int r\_no;      char name[20];      struct Student\* next;  };  void heapify(struct Student\* arr[], int n, int i) {      int largest = i;      int left = 2 \* i + 1;      int right = 2 \* i + 2;      if (left < n && arr[left]->r\_no > arr[largest]->r\_no)          largest = left;      if (right < n && arr[right]->r\_no > arr[largest]->r\_no)          largest = right;      if (largest != i) {          struct Student\* temp = arr[i];          arr[i] = arr[largest];          arr[largest] = temp;          heapify(arr, n, largest);      }  }  void heapSort(struct Student\* arr[], int n) {      for (int i = n / 2 - 1; i >= 0; i--)          heapify(arr, n, i);      for (int i = n - 1; i > 0; i--) {          struct Student\* temp = arr[0];          arr[0] = arr[i];          arr[i] = temp;          heapify(arr, i, 0);      }  }  int main() {      int n;      printf("Enter the number of students: ");      scanf("%d", &n);      struct Student\*\* studentArray = (struct Student\*\*)malloc(n \* sizeof(struct Student\*));      if (studentArray == NULL) {          printf("Memory allocation failed.\n");          return 1; // Exiting with error code      }      for (int i = 0; i < n; i++) {          studentArray[i] = (struct Student\*)malloc(sizeof(struct Student));          if (studentArray[i] == NULL) {              printf("Memory allocation failed.\n");              return 1; // Exiting with error code          }          printf("Enter roll number:");          scanf("%d", &studentArray[i]->r\_no);          printf("Enter name:");          scanf("%s", studentArray[i]->name);          studentArray[i]->next = NULL;      }      printf("\nBefore sorting:\n");      for (int i = 0; i < n; i++)          printf("%d: %s\n", studentArray[i]->r\_no, studentArray[i]->name);      heapSort(studentArray, n);      printf("\nAfter sorting:\n");      for (int i = 0; i < n; i++)          printf("%d: %s\n", studentArray[i]->r\_no, studentArray[i]->name);        for (int i = 0; i < n; i++)          free(studentArray[i]);      free(studentArray);      return 0;  } |

Output:



1. **Write a program to implement Quick sort on structure that have a member a.Rollno and b.Name.**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  struct Stud {      int rollno;      char name[20];  };  void swap(struct Stud \*a, struct Stud \*b) {      struct Stud temp = \*a;      \*a = \*b;      \*b = temp;  }  int partition(struct Stud arr[], int low, int high) {      int pivot = arr[low].rollno;      int i = low - 1;      int j = high + 1;      while (1) {          do {              i++;          } while (arr[i].rollno < pivot);          do {              j--;          } while (arr[j].rollno > pivot);          if (i >= j)              return j;          swap(&arr[i], &arr[j]);      }  }  void quickSort(struct Stud arr[], int low, int high) {      if (low < high) {          int pi = partition(arr, low, high);          quickSort(arr, low, pi - 1);          quickSort(arr, pi + 1, high);      }  }  int main() {      int n;      printf("Enter the number of students: ");      scanf("%d", &n);      struct Stud students[n];      printf("\n");      for (int i = 0; i < n; i++) {          printf("Enter roll number:");          scanf("%d", &students[i].rollno);          printf("Enter name:");          scanf("%s", students[i].name);      }      quickSort(students, 0, n - 1);      printf("\nSorted list of students:\n");      printf("Roll\tName\n");      for (int i = 0; i < n; i++) {          printf("%d\t%s\n", students[i].rollno, students[i].name);      }      return 0;  } |

Output:

