Write a program to read name, rollno, address, and phone number of each student in your class using structure. Store the information in file so that you can recover the information later. While recovering the information from the file sort the information alphabetically according to the name.

```
#include <stdio.h>
// header file for console input output
#include <string.h>
// basic data structure;
struct student
{
 char name[20];
 int roll;
 char address[20];
 char phone[10];
};
/**********
 *
 * Stores new student name and other information
 *
 * \param filename
 *********/
int store(char* filename)
{
 // opening file in appending mode
 FILE* fp = fopen(filename,"w");
 // checking if the file is available
 if (fp == NULL)
  {
   fprintf(stderr,"could not open the file");
   return -1;
  }
 // decalaring struct variable
 struct student student;
 // temp variable
 char ans;
 do
  {
   // asking user for student infromation
   printf("Enter the name of the student");
```

```
    scanf(" %s",student.name);
    printf("Enter the roll no");
    scanf(" %d",&student.roll);
    printf("Enter the address of the student");
    scanf(" %s",student.address);
    printf("Enter the phone no of the student");
    scanf(" %s",student.phone);
    // writing the student to the file
    fprintf(fp,"%s                    %d                    %s
%s\n",student.name,student.roll,student.address,student.phone);
    printf("Press y to continue\n");
    // asking if user want to continue updating the record
    scanf(" %c",&ans);
   }
 while( ans == 'y');
 fclose(fp);
 return 0;
}
/********
 *
 * mergesort student
 *
 * \param list
 * \param start
 * \param stop
 *
 ********/
int mergesort(struct student list[],int start, int stop)
{
 int middle = (start + stop)/2;
 if ((stop- start) <= 1)
  return 0;
 else
  {
    mergesort(list,start,middle);
    mergesort(list,middle,stop);
    struct student new[stop-start];
    int list1 = start;
    int list2 = middle;
    int newpos = 0;
```

```
    // joining sorting two sorted list
    while (list1 < middle || list2 < stop)
        {
         if (list1 < middle && list2 < stop)
          {
            if(strcmp(list[list1].name,list[list2].name) < 0)
                {
                 new[newpos] = list[list1];
                 list1++;
                }
            else
                {
                 new[newpos] = list[list2];
                 list2++;
                }
          }
         else if ( list1 == middle)
          {
            new[newpos] = list[list2];
            list2++;
          }
         else
          {
            new[newpos] = list[list1];
            list1++;
          }
         newpos++;
        }
    int i;
    for(i = 0, list1 = start; i < (stop-start);i++,list1++)
        {
         list[list1] = new[i];
        }

  }
}
/********
 *
 * retrives the student record from the file
 *
```

```
 * \params filename
 ********/
int retrive(char* filename)
{
 // temprorary struct
 struct student student;
 // opening the file in read mode
 FILE* fp = fopen(filename, "r");
 // checking if file was sucessfully opened
 if ( fp == NULL)
  {
   fprintf(stderr,"Could not open the file");
   return -1;
  }
 int record = 0;
 while( 1 )
  {
   record += 1;
   fscanf(fp,"%s                                %d                                %s
%s\n",student.name,&student.roll,student.address,student.phone);
    if(feof(fp))
        break;
  }
 rewind(fp);
 struct student allStudents[record];

 // getting the each input and printing them out
 int i;
 for(i = 0; i < record; i++ )
  {
   fscanf(fp,"%s                                %d                                %s
%s",allStudents[i].name,&allStudents[i].roll,allStudents[i].address,allStudents[i].phon
e);

  }
 mergesort(allStudents,0,record);

 for ( i = 0; i < record; i++)
  {
   printf("Student Name %s\n",allStudents[i].name);
```

```
    printf("Student Roll %d\n",allStudents[i].roll);
    printf("Student Address %s\n",allStudents[i].address);
    printf("Student Phone no %s\n",allStudents[i].phone);
    printf("\n");
   }
 fclose(fp);
 return 0;
}
int main()
{
 // getting user option
 char filename[50];
 printf("Enter the filename");
 scanf(" %s",filename);
 printf("What do you want to do:\n");
 printf("1. Store data\n");
 printf("2. Retrive data\n");
 int choice;
 scanf(" %d",&choice);
 switch(choice)
  {
  case 1:
   return store(filename);
  case 2:
   return retrive(filename);
  default:
   return -3;
  }
}

#include<stdio.h>//or
#include<string.h>
#include<windows.h>
typedef struct
{
  char name [20];
  int roll;
  char address[30];
  long long int number;
} std;
```

```
void sort(std *temp);
int NoOfStudents;
int main ()
{
   printf("Enter the no of students whose data is to added:\t");
   scanf("%d",&NoOfStudents);
   std bct[NoOfStudents];
   FILE *f;
   f=fopen("data.txt","w");
   int i;
   for(i=0;i<NoOfStudents;i++)
   {
      printf("Enter your name:\t");
      scanf(" %[^\n]s",bct[i].name);
      printf("\nEnter your address:\t");
      scanf(" %[^\n]s",bct[i].address);
      printf("\nEnter your roll number:\t");
      scanf("%d",&bct[i].roll);
      printf("\nEnter your phone number:\t");
      scanf("%lld",&bct[i].number);
   }
      fwrite(bct,sizeof(bct),1,f);
   fclose(f);
   f=fopen("data.txt","r");
   fread(bct,sizeof(bct),1,f);
   fclose(f);
   sort(bct);
   system("cls");
   for (i=0;i<NoOfStudents;i++)
   {
      printf("%s\n",bct[i].name);
      printf("%s\n",bct[i].address);
      printf("%d\n",bct[i].roll);
      printf("%lld\n\n",bct[i].number);
   }
}
void sort(std *temp)
{
   std t;
   for (int i=0;i<NoOfStudents-1;i++)
```

```
    {
        for (int j=i+1;j<NoOfStudents;j++)
        {
            if(strcmp(temp[i].name,temp[j].name)>0)
            {
                t=temp[i];
                temp[i]=temp[j];
                temp[j]=t;
            }
        }
    }
}
```