```c
//RAED ALSHEIKH AMIN-2528271-PRACTICAL ASSIGNMENT 1
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
#include<string.h>
#include<ctype.h>
//important notes
/*
my load function doesn't work perfectly and the data will be like that.
country1->data
.
.
country2->data
.
.
country2->data
.
.
country2->data
.
.
country7->data
.
.
country6->data
.
.
oceanx->data


*/

struct info//structure info which is gonna make the nodes that have the informations of earthquakes.
{
    int year;
    int month;
    int day;
    float magnitude;
    float lattitude;
    float logitude;
    int tsunami;
    struct info *next;//to have the next node.
};

struct country{//structure country which is gonna make the node of the country which has the name.
    char countryname[50];
    struct country *down;//it refers to the countries in the first column.
    struct info *side;//it connect each country with each info that it has.
    };
typedef struct country CO;//to make it easier to use.(co instead of all the structure)
typedef struct info inf;//to make it easier to use.(inf instead of all the structure)




CO* load_data(char[]);
void remove_earthquake(CO*,float);
void count_earthquakes(CO*,char[]);//it will take a string to search for the name
float average_magnitude(CO*,char[]);//it will search for specific country and return the avg magnitude.
```

```c

int main()
{
    FILE *infile;//declaring the varriable of the file
    char filename[20];//declaring the string of the file name.
    CO *con;//my structure pointer of country in main function
    float threshold;
    int operation;
    char countrysearch[20];
    printf("Welcome to Earthquakes Data Analysis Program \n");
    printf("Enter file name: ");
    gets(filename);//to get the file name from the user.
    while((infile=fopen(filename,"r"))==NULL)//to check if the file name is correct or not and re-ask the user to enter again.
    {
        printf("error, the file is not existed\n");
        printf("Enter file name: ");
        gets(filename);
    }
    fclose(infile);//to close the file.

    con=load_data(filename);//calling the function and assigning the strucutre that is returned to a strucutre pointer that i will use.
    printf("The earthquake characteristics is successfully loaded. \n");
    do
    {

        printf(
"-----------------------------------------------------------------\nOperations Menu\n-----------------------------------------------------------------\n1. Remove Earthquake\n2. Show Statistics\n3. Exit");
        printf("\nEnter your option: ");
        fflush(stdin);
        scanf("%d",&operation);
        if(operation==1)
        {
            printf("\nEnter a threshold value for the magnitude: ");
            scanf("%f",&threshold);//taking the value from the user to compare
            remove_earthquake(con,threshold);//calling the function.
        }
        else if(operation==2)
        {
            printf("Enter a user name: ");
            fflush(stdin);
            scanf("%s",countrysearch);//take the countryname from the user to search in the linked list.
            count_earthquakes(con,countrysearch);//calling the function.
            printf("The average of earthquake magnitude is: %.2f\n", average_magnitude(con,countrysearch));//calling and printing the avg function
        }
    }while(operation!=3);
    printf("See you, bye bye!");

}




CO* load_data(char filename[20])//the function that takes the file name and returns a structure pointer which is country.
{
    int numberoflines=0;//varriable to count the number of lines.
    char c;//character to go over the file.
    FILE *infile;//declaring the file
```

```
120      CO *countrys;//s refers to structure
121      inf *infos;//s refers to strucutre
122
123      infile=fopen(filename,"r");//opening the file to read from it
124      if(infile==NULL)//checking if the file opened correctly or not.
125      {
126          printf("couldn't proceed to the file");
127          exit(1);
128      }
129
130      while((c=fgetc(infile))!=EOF)//counting the lines in the file to make nodes as
needed.
131      {
132          if(c =='\n')
133              numberoflines++;
134      }
135      numberoflines++;
136      rewind(infile);//to restart the crosser that i can read again.
137      int i=0;
138      CO *h1=NULL;//it goes down
139      inf *h2=NULL;//right
140      CO *temp1=NULL;//it is gonna keep the list connected and go down
141      inf *temp2=NULL;//it is gonna go right
142
143
144      while(i<numberoflines)//numberoflines=number of nodes that we are going to make
145      {
146          countrys=(CO*)malloc(sizeof(CO));//creating nodes for the country columns
147          infos=(inf*)malloc(sizeof(inf));//creating nodes for the info columns
148
149          fscanf(infile,"%[^,],%d,%d,%d,%f,%f,%f,%d",countrys->countryname,&(infos->
year),&(infos->month),&(infos->day),&(infos->magnitude),&(infos->lattitude),&(infos->
logitude),&(infos->tsunami));//reading from the file.
150          i++;
151          countrys->down=NULL;//every time we make a node we set node down to null.
152          countrys->side=NULL;//we set node side that points to another node to null.
153          infos->next=NULL;//the other node which is infos points to null.
154
155
156          if(h1==NULL && h2==NULL)//it will be executed once.
157             {
158              h1=countrys;//h1 is a pointer that has the value of countrys to go over
the linked list.
159              h2=infos;//h2 is a pointer that has the value of infos to go over the
linked list
160
161              }
162          else
163          {
164              temp1=h1;
165              temp2=h2;
166              while((temp1->down)!=NULL )
167              {
168                  temp1=temp1->down;//to connect the countries together.
169                  temp2=temp2->next;//to connect the info together if we have more
than one.
170                  }
171              temp1->side=temp2;//to connect both nodes together
172              temp1->down=countrys;//to connect countrys together
173              temp2->next=infos;//to connect infos
174          }//end of else
175
176      }//end of while
177      countrys->side=infos;//countrys and infos are the tails of my program.//this
command to connect the last country with the infos.
178
```

```
179
180  return h1;
181  }
182
183  void remove_earthquake(CO* h1,float threshold)//the function to remove earthquake
which means removing the countryies.
184  {
185   CO *temp1=h1;
186   CO *temp2=h1;
187   CO *deletenodecountry;
188   int countofremovednodes=0,i=0;
189   while(temp1!=NULL)
190   {
191       if((temp1->side->magnitude<threshold) &&(temp1->side!=NULL))
192       {
193           deletenodecountry=temp1;//to specify the node to delete.
194           temp2->down=deletenodecountry->down;//to connect
195           free(deletenodecountry);//to delete the nood
196           countofremovednodes++;//number of nodes deleted.
197       }
198       if(i==1)
199       {
200          temp2=temp2->down;
201       }
202       temp1=temp1->down;
203       i=1;
204   }
205   printf("The number of removed nodes: %d\n",countofremovednodes);
206  }
207
208  void count_earthquakes(CO* h1,char countryname[20])//function to count earthquakes
for a specific country.
209  {
210  CO *temp1=h1;
211  CO *search;
212  search=(CO *)malloc(sizeof(CO));//i tried to make a node the strcmp works but it is
not.
213  strcpy(search->countryname,countryname);//copying the data from the user to a node
to use strcmp
214  int countearth=0,counttsunami=0;
215
216      while(temp1!=NULL)
217      {
218          if(strcmp(search->countryname,temp1->countryname)==0)//to compare two
strings.
219          {
220              countearth++;
221               if((temp1->side->tsunami)==1)
222          {
223              counttsunami++;
224          }
225          }
226
227
228          temp1=temp1->down;//to go over the linked list
229      }
230      printf("The number of earthquake is: %d and the Tsunami is %d\n", countearth,
counttsunami);
231  }
232  float average_magnitude(CO *h1,char countryname[20])
233  {
234      CO *temp1=h1;
235      float avg=0;
236      int count=0;
237      int valid=0;
238      while(temp1!=NULL)
```

```
239     {
240         if(strcmp(temp1->countryname,countryname)==0)
241             {
242                 avg+=temp1->side->magnitude;
243                 count++;
244                 valid=1;
245             }
246             temp1=temp1->down;
247 }
248 if(valid==0)//that means the country is not in the list
249     return 0;
250 
251 
252 return (avg/count);
253 
254 
255 }
256 
```