```
//Name: Mehmet Fatih Celik

//ID: 2385268


//For searching as stated in the assaignment, I used sequential search. Its best case O(1), worst case O(n).

//According to sample output it is O(n) as it is not ordered. There are just one loop and iterates only "n" times,

//That's why its worst case is o(n).


//For sorting as stated in the assaignment, I used insertion sort. Its best case O(n), worst case O(n^2).

//There are two loops nested, and for the average and vorst case n*n = it is o(n^2) obvious.

//for ordered inputs(for best case) it is O(n) because only the first loop iterates, second loop will not be iterated because

//the condition of tmp<a[j-1] is never going to be true.


#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct songsInfo{

	char songName[25];

	char artistName[25];

	int songYear;

};


void displaySongs(struct songsInfo *info, int num_lines);

void sortSongsYear(struct songsInfo *info, int num_lines);

void searchSong(struct songsInfo *info, int num_lines);



int main(){

	FILE *fptr;
```

```
	char ch;
	int num_lines=1, i;

	fptr=fopen("songs.txt","r");
	if (fptr == NULL){
		printf("Error occured while reading the file!");
		exit(1);
	}

	while((ch = fgetc(fptr)) != EOF ){
		if (ch == '\n')
			num_lines++;
	}

	struct songsInfo *info;
	info= (struct songsInfo*)malloc(num_lines*sizeof(struct songsInfo));  //ALLOCATION HERE
	if (info== NULL){
		printf("Error occured while allocating the memory!\n");
		exit(1);
	}

	fseek(fptr, 0L, SEEK_SET); //go to start of the file

	i=0;
	while(fscanf(fptr,
"%[^;];%[^;];%d\n",info[i].songName,info[i].artistName,&info[i].songYear)!=EOF)
		i++;

	fclose(fptr);

	char choice;
```

```
	printf("The songs.txt file has been loaded successfully!\n\n");
	do{
		fflush(stdin);
		printf("1)Display songs\n");
		printf("2)Sort songs\n");
		printf("3)Search songs\n");
		printf("4)Exit\n");
		printf("What would you like to do? ");
		scanf("%c",&choice);

		if(choice == '1'){
			printf("\n");
			displaySongs(info, num_lines);
		}
		else if (choice == '2'){
			sortSongsYear(info, num_lines);
		}
		else if (choice == '3'){
			searchSong(info, num_lines);
		}
		else{
			if(choice == '4')
				continue;
			printf("Please enter a valid number AS LISTED AT THE MENU!\n\n");
		}
	}while(choice!= '4');

	return 0;
}
```

```
void displaySongs(struct songsInfo *info, int num_lines){
	int i;
	for(i=0;i<num_lines;i++){
		printf("  %s;%s;%d\n",info[i].songName,info[i].artistName,info[i].songYear);
	}
	printf("\n");
}

void sortSongsYear(struct songsInfo *info, int num_lines){
	int i,j;
	struct songsInfo temp;
	for(i=1;i<num_lines;i++){
		temp = info[i];

		for(j=i; j>0 && temp.songYear>info[j-1].songYear;j--)
			info[j] = info[j-1];
		info[j] = temp;
	}
	printf("\n");
	displaySongs(info, num_lines);
}

void searchSong(struct songsInfo *info, int num_lines){
	int i, year, flag=0, index; //index for tracing the index of the array of the struct's year
	printf("\n\n  Enter song year: ");
	scanf("%d",&year);
	for(i=0;i<num_lines;i++){
		if (year == info[i].songYear){
			flag= 1;
			index= i;
		}
```

```
        }
        if (flag==1)
                printf("
%s;%s;%d\n\n",info[index].songName,info[index].artistName,info[index].songYear);
        else
                printf("There is not any song which is released in %d!\n\n",year);
}
```