

```

1 #include<stdio.h>
2 #include <stdlib.h>
3 #include<string.h>
4 #include<time.h>
5 /*
6 Name: Hoc Nguyen
7 Lab #8
8 Date : 03/6/2020
9 Description: This program behaves like a phonebook with extra feautres including storing entry into a file and re
trieving entries as well
10 file naming : NguyenHoc_lab8.c
11 */
12 struct friendentry{
13     char first[30];
14     char last[30];
15     char phone[30];
16     char age[3];
17 };
18 FILE *pWrite;
19 FILE *pRead;
20 void add(int *totalcontacts, struct friendentry *friends){
21     struct friendentry *temp;
22     temp = (struct friendentry *)realloc(friends, (*totalcontacts + 1) * sizeof(struct friendentry));
23     if (temp == NULL){
24         printf("\n\tSORRY PHONEBOOK IS ALREADY FULL, PLEASE DELETE A CONTACT");
25     }
26     else{
27         friends = temp;
28         ++(*totalcontacts);
29     }
30     printf("\tWHAT'S HIS\\HER FIRST NAME?:\n\t");
31     scanf("%s", friends[*totalcontacts - 1].first);
32     printf("\tWHAT'S HIS\\HER LAST NAME?:\n\t");
33     scanf("%s", friends[*totalcontacts - 1].last);
34     printf("\tWHAT'S HIS\\HER AGE?:\n\t");
35     scanf("%s", friends[*totalcontacts - 1].age);
36     printf("\tPHONE NUMBER? (Follow the exact format: ***-***-****):\n\t");
37     scanf("%s", friends[*totalcontacts - 1].phone);
38 }
39 void del(int *totalcontacts, int i, struct friendentry *friends){
40     char searchphone[20];
41     printf("\tPHONE NUMBER? (Follow the exact format: ***-***-****):\n\t");
42     scanf("%s", &searchphone);
43     int removcontact;
44     for(i = 0; i < *totalcontacts; i++){
45         if(strcmp(searchphone, friends[i].phone) == 0){
46             removcontact = i;
47             for(removcontact; removcontact <= *totalcontacts; removcontact++){
48                 strcpy(friends[removcontact].first, friends[removcontact + 1].first);
49                 strcpy(friends[removcontact].last, friends[removcontact + 1].last);
50                 strcpy(friends[removcontact].phone, friends[removcontact + 1].phone);
51                 strcpy(friends[removcontact].age, friends[removcontact + 1].age);
52             }
53             (*totalcontacts)--;
54         }
55     }
56 }
57 void show(int *totalcontacts, int i, struct friendentry *friends){
58     if(*totalcontacts > 0){
59         for(i = 0; i <= *totalcontacts - 1; i++){
60             printf("\tFIRST NAME: ||%s||\n\tLAST NAME ||%s||\n\tAGE: ||%s||\n\tPHONE #: ||%s||\n\n", friends[i].first, friends
[i].last, friends[i].age, friends[i].phone);
61         }
62     }
63     else{
64         printf("\tSORRY YOUR PHONEBOOK IS EMPTY. PLEASE ADD A FRIEND!");
65     }
66 }
67 void sort(int *totalcontacts, int i, struct friendentry *friends){
68     char tmpfirst[30];
69     char tmplast[30];
70     char tmpphone[30];

```

```

71 char tmpage[3];
72 for (i= 0; i < *totalcontacts ; i++){
73     int j = i + 1;
74     for(j; j < *totalcontacts ; j++){
75         if(strcmp(friends[i].first, friends[j].first) > 0){
76
77             strcpy(tmpfirst, friends[i].first);
78             strcpy(friends[i].first, friends[j].first);
79             strcpy(friends[j].first, tmpfirst);
80
81             strcpy(tmplast, friends[i].last);
82             strcpy(friends[i].last, friends[j].last);
83             strcpy(friends[j].last, tmplast);
84
85             strcpy(tmpphone, friends[i].phone);
86             strcpy(friends[i].phone, friends[j].phone);
87             strcpy(friends[j].phone, tmpphone);
88
89             strcpy(tmpage, friends[i].age);
90             strcpy(friends[i].age, friends[j].age);
91             strcpy(friends[j].age, tmpage);
92         }
93
94         if((strcmp(friends[i].first, friends[j].first) == 0) && (strcmp(friends[i].last, friends[j].last) > 0)){
95
96             strcpy(tmplast, friends[i].last);
97             strcpy(friends[i].last, friends[j].last);
98             strcpy(friends[j].last, tmplast);
99
100            strcpy(tmpphone, friends[i].phone);
101            strcpy(friends[i].phone, friends[j].phone);
102            strcpy(friends[j].phone, tmplast);
103
104            strcpy(tmpage, friends[i].age);
105            strcpy(friends[i].age, friends[j].age);
106            strcpy(friends[j].age, tmpage);
107        }
108    }
109 }
110 }
111 void phonebyname(int *totalcontacts, int i, struct friendentry *friends){
112     char firstphone[30];
113     char lastphone[30];
114     printf("\tWHAT'S HIS\\HER FIRST NAME?:\n\t");
115     scanf("%s", firstphone);
116     printf("\tWHAT'S HIS\\HER LAST NAME?:\n\t");
117     scanf("%s", lastphone);
118     for(i = 0; i < *totalcontacts; i++){
119         if(strcmp(friends[i].first, firstphone) == 0){
120             if(strcmp(friends[i].last, lastphone) == 0){
121                 printf("\tPHONE # of %s %s is: ||%s||", firstphone, lastphone, friends[i].phone);
122             }
123         }
124     }
125 }
126 void callrand(int x, struct friendentry *friends){
127     printf("\n\tLETS CALL ||%s %s||!: ||%s||", friends[x - 1].first, friends[x - 1].last, friends[x - 1].phone);
128 }
129 void Store(int *totalcontacts, struct friendentry *friends){
130     int select;
131     char Name[30];
132     int i;
133     printf("\n\t||1.NAME YOUR FILE||\n\t||2.USE DEFAULT NAME (phonebook.txt)||\n\t");
134     scanf("%d", &select);
135     if (select == 2)
136     {
137         pWrite = fopen("phonebook.txt", "w");
138     }
139     else
140     {
141         printf("FILE NAME?");
142         scanf("%s", Name);

```

```

143     pWrite = fopen("Name", "w");
144 }
145 if (pWrite == NULL)
146 {
147     printf("\n\tSORRY FILE CAN'T OPEN\n");
148 }
149 else
150 {
151     for (i = 0; i < *totalcontacts; i++)
152     {
153         fprintf(pWrite, "%s\t%s\t%s\t%s\n", friends[i].first, friends[i].last, friends[i].phone, friends[i]
154         i].age);
155         fclose(pWrite);
156     }
157     printf("\n\t||ENTRIES SAVED||\n\t");
158 }
159 void Retrieve(int *totalcontacts, struct friendentry *friends)
160 {
161     char rName[30];
162     int i;
163     printf("\n\t||FILE NAME?||\n\t");
164     scanf("%s", rName);
165     pRead = fopen("rName", "r");
166     if (pRead == NULL)
167     {
168         printf("\n\tSORRY FILE CAN'T OPEN\n");
169     }
170     else
171     {
172         for (i = 0; i < *totalcontacts; i++)
173         {
174             fscanf(pWrite, "%s\t%s\t%s\t%s\n", friends[i].first, friends[i].last, friends[i].phone, friends[i]
175             ].age);
176             printf("\t||%s\t%s\t%s\t%s||\n", friends[i].first, friends[i].last, friends[i].phone, friends[i].
177             age);
178         }
179     }
180 }
181 int main(){
182     struct friendentry *friends;
183     friends = (struct friendentry *)malloc(sizeof(struct friendentry));
184     int choice, totalcontacts = 0, i;
185     srand(time(NULL));
186     do{
187         printf("\n\tPLEASE SELECT THE FOLLOWING OPTIONS:\n\t||1.ADD FRIEND||\n\t||2.DELETE FRIEND||\n\t||3.SHOW PHONEB
188         OOK||\n\t||4.SORT PHONEBOOK (A - Z)||\n\t||5.FIND PHONE # BY PERSON||\n\t||6.CALL A RANDOM FRIEND||\n\t||7.STORE ALL P
189         HONEBOOK ENTRIES INTO A FILE||\n\t||8.RETRIEVE A PHONEBOOK ENTRY FROM FILE||\n\t||9.CLEAR PHONEBOOK AND EXIT||\n\t");
190         scanf("%d", &choice);
191         switch (choice){
192             case 1:
193                 add(&totalcontacts, friends);
194                 break;
195             case 2:
196                 del(&totalcontacts, i, friends);
197                 break;
198             case 3:
199                 show(&totalcontacts, i, friends);
200                 break;
201             case 4:
202                 sort(&totalcontacts, i, friends);
203                 break;
204             case 5:
205                 phonebyname(&totalcontacts, i, friends);
206                 break;
207             case 6:
208                 if(((rand()%totalcontacts) == NULL) || (totalcontacts == 0)){
209                     printf("\n\t SORRY YOU HAVEN'T ADD A FRIEND YET. PLEASE ADD BEFORE CALLING!\n");
210                 }
211                 else{
212                     int x = rand()%totalcontacts;
213                     callrand(x, friends);
214                 }
215             }
216         }
217     }

```

```
210     }
211     break;
212     case 7: //Store entries in a file
213         Store(&totalcontacts, friends);
214         break;
215
216         case 8: //Retrieve entries from file
217             Retrieve(&totalcontacts, friends);
218             break;
219     case 9:
220         printf("\t||PHONEBOOK CLEARED!||");
221         printf("\n\t||THANK YOU!GOODBYE!||\n\t");
222         free(friends);
223         break;
224     default:
225         printf("\n\tSORRY INPUT UNKNOWN PLEASE TRY AGAIN:\n");
226     }
227 }while(choice != 9);
228 return 0;
229 }
```