#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
struct Contato{  
 char nome[100];  
 int telefone;  
};  
  
int main(){  
 struct Contato \*agenda;  
 int cap;   
 int n;   
 int opc;  
 int i;  
 FILE \*fp;  
  
 fp = fopen("agenda.dat","r");  
  
 if(fp==NULL){  
 n = 0;   
 cap = 20;   
 agenda = (struct Contato \*)malloc(sizeof(struct Contato)\*cap);  
 }  
 else{  
 fread(&n, sizeof(int), 1, fp);  
 cap = n\*2;   
 agenda = (struct Contato \*)malloc(sizeof(struct Contato)\*cap);  
 fread(agenda, sizeof(struct Contato), n, fp);  
 fclose(fp);  
 }  
   
 do{  
 printf(" Menu \n");  
 printf("1) Cadastrar contato.\n");  
 printf("2) Exibir contatos.\n");  
 printf("3) Apagar.\n");  
 printf("4) Sair.\n");  
 scanf("%d",&opc);  
  
 if(opc==1){  
  
 if(n==cap){  
 cap \*= 2;  
 agenda = realloc(agenda, sizeof(struct Contato)\*cap);  
 }  
 printf("Digite o nome: ");  
 scanf(" %[^\n]",agenda[n].nome);  
 printf("Digite o telefone: ");  
 scanf("%d",&agenda[n].telefone);  
 n++;   
 }  
 else if(opc==2){  
 for(i=0; i<n; i++){  
 printf("\*\*\* Contato %d \*\*\*\n",i+1);  
 printf("Nome: %s\n",agenda[i].nome);  
 printf("Telefone: %d\n",agenda[i].telefone);  
 }  
 }  
 else if(opc==3){  
 char nome[100];  
 int j;  
 scanf("%s",&nome);  
 for(i=0; i<n; i++){  
 if (strcmp(agenda[i].nome, nome)==0){  
 for(j=i+1; j<n; j++){  
 strcpy(agenda[j-1].nome,agenda[j].nome);  
 agenda[j-1].telefone=agenda[j].telefone;  
 }  
 n--;  
 i=n;  
 }  
 }  
 }  
 }while(opc!=4);   
  
 if(n>0){  
  
 fp = fopen("agenda.dat","w");  
 fwrite(&n, sizeof(int), 1, fp);  
 fwrite(agenda, sizeof(struct Contato), n, fp);  
 fclose(fp);  
 }  
 free(agenda);  
 }

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
void limpa() {  
 printf("\e[H\e[2J");  
}  
  
void registra(char nome[15], int telefone) {  
 FILE \*fd;  
 fd=fopen("nomesagenda","a");  
 fprintf(fd, "%s\n", nome);  
 fprintf(fd, "%d\n", telefone);  
 printf("\n\tDados registrados!\n");  
 fclose(fd);  
 getchar();  
}  
  
void listalimp() {  
 FILE \*fd;  
 fd=fopen("nomesagenda","w+");  
 printf("\n\tLista limpa!\n");  
 fclose(fd);  
 getchar();  
}  
  
void pesquisa(char pesq[])  
{  
 FILE \*fd;  
 char curs[15];  
 int telefone, existe=0;  
 fd=fopen("nomesagenda","r");  
 while ((fscanf(fd,"%s", curs))!=EOF)  
 if (!strcmp(curs, pesq)) {  
 printf("\n\tNome: %s", curs);  
 fscanf(fd,"\n%d", &telefone);  
 printf("\n\tTelefone: %d\n", telefone);  
 fscanf(fd,"\n");  
 existe=1;  
 }  
 else fscanf(fd,"\n\n");  
 if (existe==0) printf("Cadastro inexistente!\n");  
 getchar();  
 fclose(fd);  
}  
  
void listar() {  
 FILE \*fd;  
 char curs[15];  
 int telefone,i=0;  
 fd=fopen("nomesagenda","r");  
 while ((fscanf(fd,"%s", curs))!=EOF) {  
 printf("\n\tNome: %s",curs);  
 fscanf(fd,"\n%d", &telefone);  
 printf("\n\tTelefone: %d\n", telefone);  
 fscanf(fd,"\n");  
 i++;  
 }  
 printf("\n\tTotal de registros: %d",i);  
}  
   
int main(void) {  
 FILE \*fd;  
 int menu,tamnome,telefone;  
 char nome[15];  
 while (menu!=0) {  
 limpa();  
 printf("\n\t Agenda de telefones. -Por hunz.\n");  
 printf("\n\t(1)\tRegistrar nome.");  
 printf("\n\t(2)\tProcurar nome.");  
 printf("\n\t(3)\tListar nomes.");  
 printf("\n\t(4)\tLimpar lista.");  
 printf("\n\t(5)\tSobre.");  
 printf("\n\n\t(0)\tSair.");  
 printf("\n\n[Menu-HZ]$ ");  
 scanf("%d",&menu);  
 getchar();  
 switch (menu) {  
 case 1: {  
 printf("\n\tEntre com o Nome: ");  
 scanf("%s",&nome);  
 getchar();  
 tamnome=strlen(nome);  
 if (tamnome>15) {  
 printf("O tamanho do nome passa de 15 letras.\n");  
 getchar(); getchar();  
 }  
 else {  
 printf("\n\tEntre com o Telefone: ");  
 scanf("%d",&telefone);  
 getchar();  
 registra(nome, telefone);  
 }  
 } break;  
 case 2: {  
 printf("\n\tEntre com o Nome: ");  
 scanf("%s",&nome);  
 getchar();  
 tamnome=strlen(nome);  
 if (tamnome>15) {  
 printf("O tamanho do nome passa de 15 letras.\n");  
 getchar(); getchar();  
 }  
 else {  
 pesquisa(nome);  
 }  
 } break;  
 case 3: {  
 listar();  
 getchar();  
 } break;  
 case 4: {  
 listalimp();  
 getchar();  
 } break;  
 case 0: break;  
 default: {  
 printf("\n\tOpcao invalida..\n");  
 getchar(); getchar();  
 };  
 }  
 }  
 return(0);  
}