

```
#include<stdio.h>
#include<stdlib.h>

typedef struct elenco
{
    char cognome[35];
    char nome[35];
    char sesso;
} Elenco;

int inserimento(Elenco **ele, int dimensione)
{int i;

    for(i=0;i<dimensione;i++)
    {

        ele[i]=malloc(sizeof(Elenco));

        fprintf(stdout,"%s","Inserisci il tuo cognome: ");
        fscanf(stdin,"%s",ele[i]->cognome);

        getchar();

        fprintf(stdout,"%s","Inserisci il tuo nome: ");
        fscanf(stdin,"%s",ele[i]->nome);

        getchar();

        fprintf(stdout,"%s","Insert a sex: ");
        fscanf(stdin,"%c",&ele[i]->sesso);

        getchar();

    }
}

int apri_fp(FILE *fp)
{
    fp = fopen("elenco.txt","w");

    if(!fp)
    {
        return 1;
    }

    return 0;
}

int libera_memoria(Elenco **ele, int dimensione)
{int i;

    for(i=0;i<dimensione;i++)
    {
        free(ele[i]);
    }

    free(ele);
}

int main(void)
{

    Elenco **ele; //array di puntatori a strutture, puntatore
                  che punta ad altri puntatori che puntano a delle strutture
    FILE *fp;
```

```
int dimensione;
int i=0;

fprintf(stdout,"%s","Quanto deve essere grande l'elenco?\n");
fprintf(stdout,"%s","Inserisci il numero: ");
fscanf(stdin,"%d",&dimensione);

    ele = malloc(sizeof(Elenco)*dimensione);           //allocazione dinamica

inserimento(ele, dimensione);

if(apri_fp(fp)) printf("Errore \n");
else{
    for(i=0;i<dimensione;i++)
    {
        fwrite(ele[i], sizeof(Elenco), 1 ,fp);
    }

    fclose(fp);

    /*LIBERA MEMORIA*/

    libera_memoria(ele, dimensione);

    //RIALLOCA STRUTTURE

    ele = malloc(sizeof(Elenco)*dimensione);

    for(i=0;i<dimensione;i++)
    {
        ele[i]=malloc(sizeof(Elenco));
    }

    //LEGGI FILE

    fp = fopen("elenco.txt","r");

    if(!fp)
    {
        exit(1);
    }

    for(i=0;i<dimensione;i++)
    {
        fread(ele[i],sizeof(Elenco),1,fp);
    }

    fclose(fp);

    //STAMPA STRUTTURE
    for(i=0;i<dimensione;i++)
    {
        fprintf(stdout,"Cognome: %s\n",ele[i]->cognome);
        fprintf(stdout,"Nome: %s\n",ele[i]->nome);
        fprintf(stdout,"sex: %c\n",ele[i]-> sesso);
    }

    libera_memoria(ele, dimensione);

}
return (0);
}
```