

```
/*  
 * Corso di Fondamenti di Informatica  
 * Esercizio:  
 * tema d'esame (2018-09-04)  
 */
```

```
/* Soluzione dell'esercizio che non fa uso della funzione rewind */
```

```
/*  
 Si consideri un file binario contenente dati di tipo Dato. Il tipo Dato Ã una struct composta da due campi: un campo intero  
 chiamato Valore e un campo carattere chiamato Chiave. Il file contiene uno o piÃ di questi dati.
```

```
Si scriva un programma C che:
```

1. apre un file binario del tipo sopra descritto, situato nella directory corrente, il cui nome Ã "Archivio";
2. legge le struct contenute nel file alla ricerca di quella (se presente) il cui campo Chiave vale '#';
3. se tale struct esiste, il programma modifica la prima delle struct contenute nel file, copiando nel suo campo Valore il campo Valore della struct il cui campo Chiave vale '#' (senza modificare le struct successive contenute nel file);
4. chiude il file.

Se il file non contiene alcuna struct il cui campo Chiave vale '#', il programma lascia inalterato il file. Se il file contiene piÃ struct il cui campo Chiave vale '#', solo la prima di esse viene presa in considerazione.

Ogniqualevolta esegue operazioni di apertura o chiusura file, il programma deve segnalare (con appositi messaggi a schermo) eventuali errori riscontrati nel loro svolgimento.

Nota: aprendo un file in modalitÃ lettura+scrittura, una operazione di scrittura eseguita in testa al file sovrascrive solo il dato situato in testa al file, senza alterare la parte restante del file.

```
*/
```

```
/* VERSIONE DEL PROGRAMMA CHE NON FA USO DELLA FUNZIONE DI LIBRERIA rewind */
```

```
#include <stdio.h>  
#define NOMEFILE "Archivio"  
#define CHIAVECERCATA '#'  
  
void CreaFileDiprova(int ChiaveCercataPresente);  
/* funzione non richiesta all'esame, utilizzata solo per debugging. Crea un  
 * file di prova del tipo richiesto dal programma. Se ChiaveCercataPresente  
 * ha valore 1, nel file vengono inseriti elementi il cui campo Chiave ha  
 * valore CHIAVECERCATA */
```

```
void StampaFile();  
/* funzione non richiesta all'esame, utilizzata solo per debugging. Stampa a  
 * schermo il contenuto di un file del tipo richiesto dal programma. */  
typedef struct  
{  
    int Valore;  
    char Chiave;  
} Dato;  
  
typedef enum {false,true} boolean;  
  
int main()  
{  
    FILE* pFile;  
    /* Io stream da elaborare */  
    Dato DatoLetto;  
    /* dato prelevato dallo stream */  
    boolean ChiaveTrovata;  
    /* vale true se e' stato trovato un elemento con il valore cercato del campo Chiave */  
    int NuovoValore;  
    /* nuovo valore da inserire nel campo Valore del primo dato nel file */  
    int EsitoChiusura;  
    /* valore restituito da fclose */  
  
    CreaFileDiprova(1);  
    /* non richiesta all'esame, utilizzata per debugging */  
    StampaFile();  
    /* non richiesta all'esame, utilizzata per debugging */  
  
    pFile = fopen(NOMEFILE, "rb");  
  
    if ( NULL == pFile )  
    {  
        printf("\nErrore in fase di apertura del file '%s'\n", NOMEFILE);  
    }  
    else  
    {  
        printf("\nIl file '%s' e' stato correttamente aperto in lettura.\n", NOMEFILE);  
  
        fread(&DatoLetto, sizeof(Dato), 1, pFile);  
    }  
}
```

```
/* questa lettura serve a superare il primo elemento; il valore letto non viene utilizzato */
ChiaveTrovata = false;

while ((false == ChiaveTrovata) && (1 == fread(&Datoletto, sizeof(Dato), 1, pFile)))
/* esaminiamo gli altri dati contenuti nel file fino a che il file termina
 * oppure fino a che troviamo un dato il cui valore del campo Chiave e'
 * quello cercato */
{
    if (CHIAVECERCATA == Datoletto.Chiave)
    {
        ChiaveTrovata = true;
        NuovovValore = Datoletto.Valore;
    }
}

if (true == ChiaveTrovata)
/* se la lettura e' terminata perche' l'elemento cercato e' stato trovato */
{
    EsitoChiusura = fclose(pFile);

    if (0 != EsitoChiusura)
    {
        printf("\nErrore in fase di chiusura del file '%s'\n", NOMEFILE);
    }
    else
    {
        printf("\nIl file '%s' e' stato correttamente chiuso.\n", NOMEFILE);
    }

    pFile = fopen(NOMEFILE, "rb");

    if ( NULL == pFile )
    {
        printf("\nErrore in fase di apertura del file '%s'\n", NOMEFILE);
    }
    else
    {
        printf("\nIl file '%s' e' stato correttamente riaperto in lettura.\n", NOMEFILE);
    }
}
```

```
fread(&DatoLetto, sizeof(Dato), 1, pFile);
DatoLetto.Valore = NuovoValore;

EsitoChiusura = fclose(pFile);

if (0 != EsitoChiusura)
{
    printf("\nErrore in fase di chiusura del file '%s'\n", NOMEFILE);
}
else
{
    printf("\nIl file '%s' e' stato correttamente chiuso.\n", NOMEFILE);
}

pFile = fopen(NOMEFILE, "rb+");

if ( NULL == pFile )
{
    printf("\nErrore in fase di apertura del file '%s'\n", NOMEFILE);
}
else
{
    printf("\nIl file '%s' e' stato correttamente riaperto in lettura+scrittura.\n", NOMEFILE);
}

fwrite(&DatoLetto, sizeof(Dato), 1, pFile);
/* questa operazione di scrittura sovrascrive il primo dato nel file
 * senza alterare i dati successivi */
}

EsitoChiusura = fclose(pFile);

if (0 != EsitoChiusura)
{
    printf("\nErrore in fase di chiusura del file '%s'\n", NOMEFILE);
}
else
{
    printf("\nIl file '%s' e' stato correttamente chiuso.\n", NOMEFILE);
}
}
```

```
StampaFile();
/* non richiesta all'esame, utilizzata per debugging */
return(0);
}

/*****
* tutto quanto segue non era richiesto all'esame
* ed e' presente soltanto per ragioni di debugging
*****/

void CreaFileDiprova(int ChiaveCercataPresente)
{
    FILE* pFile;
    /* stream associato al file da creare */
    Dato Nuovodato;
    /* dato da inserire nel file */
    int Contatore;
    /* usato per generare dati di prova */

    pFile = fopen(NOMEFILE, "wb");

    for (Contatore = 1; Contatore < 25; ++Contatore )
    {
        Nuovodato.Valore = Contatore;
        if ((0 == Contatore%7) && (1 == ChiaveCercataPresente))
        {
            Nuovodato.Chiave = CHIAVECERCATA;
        }
        else
        {
            Nuovodato.Chiave = 'a' + Contatore - 1;
        }

        fwrite(&Nuovodato, sizeof(Dato), 1, pFile);
    }

    fclose(pFile);
}
```

```
void StampaFile()  
{  
    FILE* pFile;  
    /* stream associato al file da esaminare */  
    Dato DatoLetto;  
    /* dato prelevato dal file */  
    printf("\nContenuto del file '%s':\n", NOMEFILE);  
    pFile = fopen(NOMEFILE, "rb");  
    while (1 == fread(&DatoLetto, sizeof(Dato), 1, pFile))  
        /* fino a che esistono dati da leggere nel file */  
        {  
            printf("%d,%c\n", DatoLetto.Valore, DatoLetto.Chiave);  
        }  
    fclose(pFile);  
}
```