by Deitel & Associates

CREARE E SCRIVERE UN FILE SEQUENZIALE

```c
/* Create a sequential file */
#include <stdio.h>
int main()
{
   int account;     /* account number */
   char name[ 30 ]; /* account name */
   double balance;  /* account balance */

   FILE *cfPtr;     /* cfPtr = clients.dat file pointer */

   /* fopen opens file. Exit program if unable to create file  */
   if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL ) {
      printf( "File could not be opened\n" );
   } /* end if */
   else {
      printf( "Enter the account, name, and balance.\n" );
      printf( "Enter EOF to end input.\n" );
      printf( "? " );
      scanf( "%d%s%lf", &account, name, &balance );
      /* write account, name and balance into file with fprintf */
      while ( !feof( stdin ) ) {
         fprintf( cfPtr, "%d %s %.2f\n", account, name, balance );
         printf( "? " );
         scanf( "%d%s%lf", &account, name, &balance );
      } /* end while */
      fclose( cfPtr ); /* fclose closes file */
   } /* end else */
   return 0; /* indicates successful termination */
} /* end main */
```

by Deitel & Associates

LEGGERE E STAMPARE UN FILE SEQUENZIALE

```c
/* Reading and printing a sequential file */

#include <stdio.h>

int main()
{
   int account;     /* account number */
   char name[ 30 ]; /* account name */
   double balance;  /* account balance */

   FILE *cfPtr;     /* cfPtr = clients.dat file pointer */

   /* fopen opens file; exits program if file cannot be opened */
   if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL ) {
      printf( "File could not be opened\n" );
   } /* end if */
   else { /* read account, name and balance from file */
      printf( "%-10s%-13s%s\n", "Account", "Name", "Balance" );
      fscanf( cfPtr, "%d%s%lf", &account, name, &balance );

      /* while not end of file */
      while ( !feof( cfPtr ) ) {
         printf( "%-10d%-13s%7.2f\n", account, name, balance );
         fscanf( cfPtr, "%d%s%lf", &account, name, &balance );
      } /* end while */

      fclose( cfPtr ); /* fclose closes the file */
   } /* end else */

   return 0; /* indicates successful termination */

} /* end main */
```

by Deitel & Associates

ELABORARE I DATI CONTENUTI IN UN FILE SEQUENZIALE

(parte 1 di 4)

```
1  /* Credit inquiry program */
3  #include <stdio.h>
5  /* function main begins program execution */
6  int main()
7  {
8     int request;     /* request number */
9     int account;     /* account number */
10    double balance;  /* account balance */
11    char name[ 30 ]; /* account name */
12    FILE *cfPtr;     /* clients.dat file pointer */
13
14    /* fopen opens the file; exits program if file cannot be opened */
15    if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL ) {
16       printf( "File could not be opened\n" );
17    } /* end if */
18    else {
20       /* display request options */
21       printf( "Enter request\n"
22               " 1 - List accounts with zero balances\n"
23               " 2 - List accounts with credit balances\n"
24               " 3 - List accounts with debit balances\n"
25               " 4 - End of run\n? " );
```

by Deitel & Associates

ELABORARE I DATI CONTENUTI IN UN FILE SEQUENZIALE

(parte 2 di 4)

```
26          scanf( "%d", &request );
27
28       /* process user's request */
29       while ( request != 4 ) {
30
31          /* read account, name and balance from file */
32          fscanf( cfPtr, "%d%s%lf", &account, name, &balance );
33
34          switch ( request ) {
35
36             case 1:
37                printf( "\nAccounts with zero balances:\n" );
38
39                /* read file contents (until eof) */
40                while ( !feof( cfPtr ) ) {
41
42                   if ( balance == 0 ) {
43                      printf( "%-10d%-13s%7.2f\n",
44                              account, name, balance );
45                   } /* end if */
46
47                   /* read account, name and balance from file */
48                   fscanf( cfPtr, "%d%s%lf",
49                           &account, name, &balance );
50                } /* end while */
52                break;
```

by Deitel & Associates

ELABORARE I DATI CONTENUTI IN UN FILE SEQUENZIALE

(parte 3 di 4)

```
54             case 2:
55                printf( "\nAccounts with credit balances:\n" );
56
57                /* read file contents (until eof) */
58                while ( !feof( cfPtr ) ) {
59
60                   if ( balance < 0 ) {
61                      printf( "%-10d%-13s%7.2f\n",
62                              account, name, balance );
63                   } /* end if */
64
65                   /* read account, name and balance from file */
66                   fscanf( cfPtr, "%d%s%lf",
67                           &account, name, &balance );
68                } /* end while */
69
70                break;
71
```

by Deitel & Associates

ELABORARE I DATI CONTENUTI IN UN FILE SEQUENZIALE

(parte 4 di 4)

```
72             case 3:
73                printf( "\nAccounts with debit balances:\n" );
74
75                /* read file contents (until eof) */
76                while ( !feof( cfPtr ) ) {
78                   if ( balance > 0 ) {
79                      printf( "%-10d%-13s%7.2f\n",
80                              account, name, balance );
81                   } /* end if */
82
83                   /* read account, name and balance from file */
84                   fscanf( cfPtr, "%d%s%lf",
85                           &account, name, &balance );
86                } /* end while */
87
88                break;
89
90          } /* end switch */
91
92          rewind( cfPtr ); /* return cfPtr to beginning of file */
94          printf( "\n? " );
95          scanf( "%d", &request );
96       } /* end while */
98       printf( "End of run.\n" );
99       fclose( cfPtr ); /* fclose closes the file */
100   } /* end else */
102   return 0; /* indicates successful termination */
104 } /* end main */
```

by Deitel & Associates

CREARE E SCRIVERE SEQUENZIALMENTE UN FILE AD ACCESSO CASUALE

(Parte 1 di 2)

```
1  /* Creating a randomly accessed file sequentially */
3  #include <stdio.h>
5  /* clientData structure definition */
6  struct clientData {
7     int acctNum;          /* account number */
8     char lastName[ 15 ];  /* account last name */
9     char firstName[ 10 ]; /* account first name */
10    double balance;       /* account balance */
11 }; /* end structure clientData */
12
13 int main()
14 {
15    int i; /* counter */
17    /* create clientData with no information */
18    struct clientData blankClient = { 0, "", "", 0.0 };
19
20    FILE *cfPtr; /* credit.dat file pointer */
21
```

by Deitel & Associates

CREARE E SCRIVERE SEQUENZIALMENTE UN FILE AD ACCESSO CASUALE

(Parte 2 di 2)

```
22    /* fopen opens the file; exits if file cannot be opened */
23    if ( ( cfPtr = fopen( "credit.dat", "wb" ) ) == NULL ) {
24       printf( "File could not be opened.\n" );
25    } /* end if */
26    else {
27
28       /* output 100 blank records to file */
29       for ( i = 1; i <= 100; i++ ) {
30          fwrite( &blankClient, sizeof( struct clientData ), 1, cfPtr );
31       } /* end for */
32
33       fclose ( cfPtr ); /* fclose closes the file */
34    } /* end else */
35
36    return 0; /* indicates successful termination */
37
38 } /* end main */
```

by Deitel & Associates

SCRIVERE CASUALMENTE IN UN FILE AD ACCESSO CASUALE

(Parte 1 di 3)

```
1  /* 2     Writing to a random access file */
3  #include <stdio.h>
5  /* clientData structure definition */
6  struct clientData {
7     int acctNum;           /* account number */
8     char lastName[ 15 ];  /* account last name */
9     char firstName[ 10 ]; /* account first name */
10     double balance;       /* account balance */
11 }; /* end structure clientData */
12
13 int main()
14 {
15    FILE *cfPtr; /* credit.dat file pointer */
17    /* create clientData with no information */
18    struct clientData client = { 0, "", "", 0.0 };
19
20    /* fopen opens the file; exits if file cannot be opened */
21    if ( ( cfPtr = fopen( "credit.dat", "rb+" ) ) == NULL ) {
22       printf( "File could not be opened.\n" );
23    } /* end if */
```

by Deitel & Associates

SCRIVERE CASUALMENTE IN UN FILE AD ACCESSO CASUALE

(Parte 2 di 3)

```
24    else {
26       /* require user to specify account number */
27       printf( "Enter account number"
28               " ( 1 to 100, 0 to end input )\n? " );
29       scanf( "%d", &client.acctNum );
30
31       /* user enters information, which is copied into file */
32       while ( client.acctNum != 0 ) {
33
34          /* user enters last name, first name and balance */
35          printf( "Enter lastname, firstname, balance\n? " );
36
37          /* set record lastName, firstName and balance value */
38          fscanf( stdin, "%s%s%lf", client.lastName,
39                  client.firstName, &client.balance );
40
```

by Deitel & Associates

SCRIVERE CASUALMENTE IN UN FILE AD ACCESSO CASUALE

(Parte 3 di 3)

```
41          /* seek position in file of user-specified record */
42          fseek( cfPtr, ( client.acctNum - 1 ) *
43                 sizeof( struct clientData ), SEEK_SET );
44
45          /* write user-specified information in file */
46          fwrite( &client, sizeof( struct clientData ), 1, cfPtr );
47
48          /* enable user to specify another account number */
49          printf( "Enter account number\n? " );
50          scanf( "%d", &client.acctNum );
51       } /* end while */
52
53       fclose( cfPtr ); /* fclose closes the file */
54    } /* end else */
55
56    return 0; /* indicates successful termination */
57
58 } /* end main */
```

ERRORI TIPICI:

Aprire in scrittura (“w”) un file già esistente senza avere intenzione di eliminare il contenuto precedente.

Aprire in scrittura (“w”) un file per aggiornarne il contenuto (“r+”), provocando l’indesiderata eliminazione del contenuto precedente.

Aprire in lettura (“r”) un file inesistente.

Dimenticare di aprire il file prima di farvi riferimento.

Usare il puntatore sbagliato per fare riferimento ad un file.

Non chiudere un file appena non se ne farà più uso.

MODALITÀ DI APERTURA DI UN FILE (una lista più dettagliata)

Mode	Description
`r`	Open a file for reading.
`w`	Create a file for writing. If the file already exists, discard the current contents.
`a`	Append; open or create a file for writing at end of file.
`r+`	Open a file for update (reading and writing).
`w+`	Create a file for update. If the file already exists, discard the current contents.
`a+`	Append; open or create a file for update; writing is done at the end of the file.
`rb`	Open a file for reading in binary mode.
`wb`	Create a file for writing in binary mode. If the file already exists, discard the current contents.
`ab`	Append; open or create a file for writing at end of file in binary mode.
`rb+`	Open a file for update (reading and writing) in binary mode.
`wb+`	Create a file for update in binary mode. If the file already exists, discard the current contents.
`ab+`	Append; open or create a file for update in binary mode; writing is done at the end of the file.