FILE MANIPULATION USING C LANGUAGE

Record Definition

```
typedef struct EMPREC {
        long int ssn;       //the EMP's social security Numb.
        char     name;      //the EMP's name
        };
```

Opening/Closing a file

```
        myf= fopen ("d:\\temp\\junk.out", "rb");
        …
        fclose(myf);
```

Seek and Read

```
        position = 0;   //fetch first record
        code=  fseek(myf, position*sizeof(e1), 0);
        code=  fread(&e1,sizeof(e1),1,myf);
```

Seek and Write

```
        e1.ssn= 0;    e1.name= 'X';
        position = 1;   //fetch second record
        code=  fseek(myf,  position*sizeof(e1), 0 );
        code=  fwrite (&e1, sizeof(e1), 1, myf);
```

FILE MANIPULATION USING C LANGUAGE

Syntax

#include <stdio.h>

FILE ***fopen**(const char ***filename**, const char ***mode**);

Description

Opens a stream.

- fopen opens the file named by *filename* and associates a stream with it. fopen returns a pointer to be used to identify the stream in subsequent operations.
- The *mode* string used in calls to fopen is one of the following values:

Value	Description
r	Open for reading only.
w	Create for writing. If a file by that name already exists, it will be overwritten.
a	Append; open for writing at end-of-file or create for writing if the file does not exist.
r+	Open an existing file for update (reading and writing).
w+	Create a new file for update (reading and writing). If a file by that name already exists, it will be overwritten.
a+	Open for append (or create if the file does not exist) for update at the end of the file.

Syntax

#include <stdio.h>

int **fseek**(FILE *stream, long offset, int whence);

Description: Repositions a file pointer on a stream.

- fseek sets the file pointer associated with stream to a new position that is *offset* bytes from the file location given by *whence*.
- For text mode streams *offset* should be 0 or a value returned by ftell.
- *whence* must be one of the values 0, 1, or 2 which represent three symbolic constants (defined in stdio.h) as follows:

Constant	Whence	File location
SEEK_SET	0	File beginning
SEEK_CUR	1	Current file pointer position
SEEK_END	2	End-of-file

- fseek discards any character pushed back using ungetc.
- fseek is used with stream I/O; for file handle I/O use lseek.
- After fseek the next operation on an update file can be either input or output.

Return Value:
fseek returns 0 if the pointer is successfully moved and nonzero on failure.

FILE MANIPULATION USING C LANGUAGE

Syntax

#include <stdio.h>

size_t **fread**(void ***ptr**, size_t **size**, size_t **n**, FILE ***stream**);

Description
Reads data from a stream.

- fread reads *n* items of data each of length *size* bytes from the given input *stream* into a block pointed to by *ptr*.
- The total number of bytes read is (n * size).

Return Value
- *On success* fread returns the number of items (not bytes) actually read.
- On *end-of-file* or error it returns a short count (possibly 0).

FILE MANIPULATION USING C LANGUAGE

#include <stdio.h>

size_t **fwrite**(const void ***ptr**, size_t **size**, size_t **n**, FILE ***stream**);

Description

Writes to a stream.

- fwrite appends *n* items of data each of length *size* bytes to the given output file.
- The data written begins at *ptr*.
- The total number of bytes written is (n * size).
- *ptr* in the declarations is a pointer to any object.

Return Value

- On successful completion fwrite returns the number of items (not bytes) actually written.
- On error it returns a short count (possibly 0).

FILE MANIPULATION USING C LANGUAGE

PART1. Making a RANDOM file

```
//create a small disk file to be processed in RANDOM mode
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#define  TOTALREC 10

typedef struct EMPREC {
        long int ssn;     //the EMP's social security Numb.
        char     name;    //the EMP's name
        };

EMPREC e1;
int    code=1, i;
char   value= 'A';
FILE * myf;

//-----------------------------------------------------------------------
void showmyf()
{
//show what is inside myf
printf("\n myf file\n");
myf= fopen ("d:\\temp\\junk.out", "rb");

for(i=0; i<=TOTALREC; i++ ) {
        code= fseek(myf,i*sizeof(e1),0);
        code= fread(&e1,sizeof(e1),1,myf);
        printf ("\n pos: %3d  %4ld  ", i, e1.ssn );
        printf ("%c",  e1.name );
        }
fclose(myf);
}
//-------------------------------------------------------
main()
{
clrscr();
myf= fopen ("d:\\temp\\junk.out", "wb");
if (myf == NULL ) {
        printf ("\nFile not created!!");
        return(1);
        };

//write one dummy record (at position zero - to be updated later)
  e1.ssn= 0; e1.name= '@';
  code= fwrite (&e1, sizeof(e1), 1, myf);

//proceed with the rest of the records
value= 'A';
for (i=1; i<= TOTALREC; i++) {
        //assemble the record here using the values 1 A,  2 B,  3  C ....
        e1.ssn= i;              e1.name= value++;
        code= fwrite (&e1, sizeof(e1), 1, myf);
        };

//resets pointer to the beginnibg of the file to
//update the total count of records in the file.
        rewind(myf);
        //e1.ssn holds the total of rec. (this is control data)
        e1.ssn= 10L; e1.name= '*';
        //write control info
        code= fwrite (&e1, sizeof(e1), 1, myf);
        fclose(myf);
showmyf();
}
```

```
(Inactive D:\TEMP1\RAN01.EXE)

myf file

pos:   0    10  *
pos:   1     1  A
pos:   2     2  B
pos:   3     3  C
pos:   4     4  D
pos:   5     5  E
pos:   6     6  F
pos:   7     7  G
pos:   8     8  H
pos:   9     9  I
pos:  10    10  J
```

FILE MANIPULATION USING C LANGUAGE

Part2. Reading and Writing from a RANDOM file

```
//open a random file and show its records
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream.h>
#include <conio.h>

typedef struct EMPREC {
        long int ssn;
        char     name;
        };


main()
{

EMPREC e1;
char   c;
int    code=1, i, value= 'A', numberofrec;
FILE * myf;
clrscr();

myf= fopen("d:\\temp\\junk.out", "r+");
if (myf == NULL) {
        cout << "\n File NOT found ";
        return(2);
        };

//read control info from record zero
code= fseek (myf, 0*sizeof(e1), 0);
code= fread(&e1,sizeof(e1),1,myf);
numberofrec= e1.ssn;   //the ssn field of rec. zero is the
                       //total number of data records in myf

cout<< "\nTotal number of records: "<< numberofrec;

/*
* NOTE: finding out how many records are there could also be done with:
*   code= fseek(myf,0,SEEK_END);
*   i=  (ftell(myf) / sizeof(e1)) -1;
*/

//printing the file in reverse
for (i=numberofrec; i > -1; i-=3)  {
        fseek (myf, i*sizeof(e1), 0);
        code= fread(&e1,sizeof(e1),1,myf);
        cout <<  "\nRandom-read " << i<< "  "<< e1.ssn <<  "  " << e1.name;
        };

//printing sequentially from the beginning
cout << "\n\n";

i=0;
code= fseek (myf, i*sizeof(e1), 0);
do      {
        code= fseek (myf, i*sizeof(e1), 0);
        code= fread(&e1,sizeof(e1),1,myf);
        if (code==1) {
          cout <<  "\nSeq-reading " << i++ << "  "
                        << e1.ssn << "  " << e1.name;
          }
        } while ( code == 1 );  //until no more data is found

}
```

```
(Inactive D:\TEMP1\RAN02.EXE)

Total number of records: 10
Random-read 10  10  J
Random-read 7   7   G
Random-read 4   4   D
Random-read 1   1   A


Seq-reading 0   10  *
Seq-reading 1   1   A
Seq-reading 2   2   B
Seq-reading 3   3   C
Seq-reading 4   4   D
Seq-reading 5   5   E
Seq-reading 6   6   F
Seq-reading 7   7   G
Seq-reading 8   8   H
Seq-reading 9   9   I
Seq-reading 10  10  J
```

Part3. Updating a RANDOM file

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream.h>
#include <conio.h>
#include <io.h>


typedef struct EMPREC {
            long int ssn;
            char     name;
            };
FILE *  myf;   //data file globally defined;


//-------------------------------------------------------
int rewrite (int pos, int newssn, char newname)
{EMPREC e1; int code;
            //re-position file pointer and re-write the record
            //with the new data
            e1.ssn = newssn;
            e1.name= newname;
            code= fseek(myf,pos*sizeof(e1),0);
            code= fwrite(&e1,sizeof(e1),1,myf);
            if (code!=1) {printf(" Problems re-writting"); };
            return(code);
}
//-------------------------------------------------------
int reread (int recnumber, EMPREC* e1)
{int code;
            code= fseek (myf, recnumber*sizeof(*e1), 0);
            code= fread(e1,sizeof(*e1),1,myf);
            return(code); //code = 1 indicates record was found
}
//-------------------------------------------------------
int add_dummy_rec (int numberofrec, int newposition)
{ EMPREC e1; int pos; int code;
 //new dummy rec. to be added (padding) between last rec
 //and new one to be inserted
 e1.ssn =-1; e1.name='#'; //a dummy record: <-1,#>
 code= fseek(myf,numberofrec*sizeof(e1),0);
 for(pos=numberofrec; pos<=newposition; pos++) {
            code= fwrite(&e1,sizeof(e1),1,myf);
            }
 return(newposition - numberofrec + 1);
}
// -------------------------------------------------------
main()
{

EMPREC e1;
long    numberofrec; //number of records in the file
int     code=1, i, yesno= 'y', newdummies, pos;
long    fsize;
char    newname; int newssn;

clrscr();

//try to open the file for input/output (rb+ mode)
myf= fopen("d:\\temp\\junk.out", "rb+");
if (myf == NULL) {
            cout << "\n File NOT found ";
            return(2);
            };
```

```
(Inactive D:\TEMP1\RAN03.EXE)

Look up into file positions 0 - 10
Enter index: [finish ^Z ]:  2

From file (pos - rec): 2  2  B
Do you want to change the record? [y/n]: y

Enter new ssn:  777

Enter new name: X

Enter index: [finish ^Z ]:
   0   10 B
   1    1 A
   2  777 X
   3    3 C
   4    4 D
   5    5 E
   6    6 F
   7    7 G
   8    8 H
   9    9 I
  10    0 A
```

```
rewind(myf);
code= fread(&e1,sizeof(e1),1,myf);
numberofrec= e1.ssn;   //this number is the total count of rec

//ask the user to provide a position between 0 to
//numberofrec, display (if possible) the rec. and
//ask if changes are to be applied
printf("\nLook up into file positions 0 - %d", numberofrec);
while ( printf("\nEnter index: [finish ^Z ]:  "),
                        ((code =scanf("%d", &i)) != EOF) )
            {
            //see if such a record exists
            code= reread(i, &e1);

            if (code!=1)
                        {
                         printf("\nProblems reading - record %d not available", i);
                         printf("\nAdd new record at this postion? [y/n]: "); fflush(stdin);
                         scanf ("%c",&yesno);
                         if (yesno == 'y')
                                    {
                                    //append dummy recs <-1,#> at the end of the file
                                    newdummies= add_dummy_rec(numberofrec, i);
                                    //put special marks in the new last record
                                    code= rewrite (i, 0, '*');
                                    numberofrec= numberofrec + newdummies;
                                    }

                        }
            else {
                         cout <<  "\nFrom file (pos - rec): "<< i <<"  "
                                    << e1.ssn <<  "  " << e1.name;
                         printf("\nDo you want to change the record? [y/n]: "); fflush(stdin);
                         scanf ("%c",&yesno);
                         if (yesno == 'y')
                                    {
                                    printf("\nEnter new ssn:  "); scanf("%d", &newssn);
                                    printf("\nEnter new name: "); fflush(stdin);

                                   scanf("%c", &newname);
                                    //modify record in position i with new data
                                    code= rewrite(i, newssn, newname);
                                    }
                         }
            }
//updating record zero (control data)
e1.ssn= numberofrec;
code= fseek(myf,0,0); // same as rewind(myf);
code= fwrite(&e1,sizeof(e1),1,myf);

//show data in the file
rewind(myf);
for (i=0;i<=numberofrec;i++) {
            code= fseek(myf,i*sizeof(e1),0);
            code= fread(&e1,sizeof(e1),1,myf);
            printf("\n %3d %4ld %c", i,e1.ssn, e1.name);
            }
fclose(myf);

return 0;
}
```