Position in file

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

void showpos(FILE *stream);

int main(void)
{
   FILE *stream;
   fpos_t filepos;

   /* open a file for update */
   stream = fopen("DUMMY.FIL", "w+");

   /* save the file pointer position */
   fgetpos(stream, &filepos);

   /* write some data to the file */
   fprintf(stream, "This is a test");

   /* show the current file position */
   showpos(stream);

   /* set a new file position, display it */
   if (fsetpos(stream, &filepos) == 0)
     showpos(stream);
   else
   {
      fprintf(stderr, "Error setting file pointer.\n");
      exit(1);
   }

   /* close the file */
   fclose(stream);
   return 0;
}

void showpos(FILE *stream)
{
   fpos_t pos;

   /* display the current file pointer
      position of a stream */
   fgetpos(stream, &pos);
   printf("File position: %ld\n", pos);
}
```

Binary read from a file

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

int main(void)
{
   FILE *stream;
   char msg[] = "this is a test";
   char buf[20];

   if ((stream = fopen("DUMMY.FIL", "w+"))
       == NULL)
   {
      fprintf(stderr, "Cannot open output file.\n");
      return 1;
   }

   /* write some data to the file */
   fwrite(msg, strlen(msg)+1, 1, stream);

   /* seek to the beginning of the file */
   fseek(stream, SEEK_SET, 0);

   /* read the data and display it */
   fread(buf, strlen(msg)+1, 1, stream);
   printf("%s\n", buf);
   fclose(stream);
   return 0;
}
```

Binary writing to a file

```
#include <stdio.h>

struct mystruct
{
  int i;
  char ch;
};

int main(void)
{
   FILE *stream;
   struct mystruct s;

   if ((stream = fopen("TEST.$$$", "wb")) == NULL)
            /* open file TEST.$$$ */
   {
      fprintf(stderr, "Cannot open output file.\n");
      return 1;
   }
   s.i = 0;
   s.ch = 'A';
   fwrite(&s, sizeof(s), 1, stream);
         /* write struct s to file */
   fclose(stream); /* close file */
   return 0;
}
```

Reopenning of a file

```
#include <stdio.h>

int main(void)
{
   /* redirect standard output to a file */
   if (freopen("OUTPUT.FIL", "w", stdout)
       == NULL)
      fprintf(stderr, "error redirecting stdout\n");

   /* this output will go to a file */
   printf("This will go into a file.");

   /* close the standard output stream */
   fclose(stdout);

   return 0;
}
```

File reading

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define DL_NAZW 20
     /*  maximal name length  */
#define LBDZ_MAX 15
     /*  maximal number of children  */
#define DL_NAZPLIKU 20
     /*  maximal length of the file name  */

void main()
{
   char nazwapliku[DL_NAZPLIKU+1];
      /*  name of the created file  */
   FILE *wy;
      /*  output file stream  */
   struct {
      char nazw[DL_NAZW+1];
      /*  description of one record of the file */
      int wiek;
      int  lb_dz;
      int wiek_dz[LBDZ_MAX];
   } blok;
   int i;

   /*   Open the file  */
   /*   Sometimes opening mode  w  instead of  wb  */
   printf("Enter the name of the file to be "
          "created: ");
   gets(nazwapliku);
   if ((wy = fopen(nazwapliku, "wb")) == NULL) {
      printf("***** Opening error - "
             "exit the program!\n");
      exit(-1);
   }

   /*  Reading data from the standard input and
       writing it to the file  */
   printf("-----At the end enter 'empty name' ----\n");
   do {
      printf("Family name        : ");
      gets(blok.nazw);
      if (strlen(blok.nazw) == 0) break;
      printf("age                 : ");
      scanf("%d",&blok.wiek);
      printf("Number of children : ");
      scanf("%d",&blok.lb_dz);
      for (i=0; i<blok.lb_dz && i<LBDZ_MAX; i++) {
       printf("Age of child number %d : ",i+1);
       scanf("%d", &blok.wiek_dz[i]);
      }
      getchar();        /*  jump over \n  */
      printf("\n");
      fwrite(&blok, sizeof(blok), 1, wy);
          /*  writing to the output file */
   } while (1);

   /*  End  */
   fclose(wy);
   printf("\n --------------- END --------------- \n");
}
```

Writing into a file

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define DL_NAZW 20
      /*  maximal name length  */
#define LBDZ_MAX 15
      /*  maximal number of children  */
#define DL_NAZPLIKU 20
      /*  maximal length of the file name  */

void main()
{
   char nazwapliku[DL_NAZPLIKU+1];
            /*  name of the created file  */
   FILE *wy;
            /*  output file stream  */
   struct {
      char nazw[DL_NAZW+1];
      /*  description of one record of the file */
      int wiek;
      int  lb_dz;
      int wiek_dz[LBDZ_MAX];
   } blok;
   int i;

   /*   Open the file  */
   /*   Sometimes opening mode  w  instead of  wb */
   printf("Enter the name of the file to be "
          "created: ");
   gets(nazwapliku);
   if ((wy = fopen(nazwapliku, "wb")) == NULL) {
      printf("***** Opening error -"
             " exit the program!\n");
      exit(-1);
   }

   /*  Reading data from the standard input and
       writing it to the file  */
   printf("-----At the end enter 'empty name' ----\n");
   do {
      printf("Family name        : ");
      gets(blok.nazw);
      if (strlen(blok.nazw) == 0) break;
      printf("age                 : ");
      scanf("%d",&blok.wiek);
      printf("Number of children : ");
      scanf("%d",&blok.lb_dz);
      for (i=0; i<blok.lb_dz && i<LBDZ_MAX; i++) {
       printf("Age of child number %d : ",i+1);
       scanf("%d", &blok.wiek_dz[i]);
      }
      getchar();        /*  jump over \n  */
      printf("\n");
      fwrite(&blok, sizeof(blok), 1, wy);
            /*  writing to the output file */
   } while (1);

   /*  End  */
   fclose(wy);
   printf("\n --------------- END --------------- \n");
}
```

File correction

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

#define DL_NAZW 20
      /*  maximal name length  */
#define LBDZ_MAX 15
      /*  maximal number of children  */
#define DL_NAZPLIKU 20
      /*  maximal length of the file name  */

struct rek {
   char nazw[DL_NAZW+1];
      /*  description of one record of the file */
   int wiek;
   int  lb_dz;
   int wiek_dz[LBDZ_MAX];
} ;

void main()
```

```
{
   void wyswietl(struct rek *, int);
         /* function for display  */
   void zmien(struct rek * );
         /* change in the file content  */
   char nazwapliku[DL_NAZPLIKU+1];
         /*  name of the file  */
   FILE *plik;
         /*  file description    */
   struct rek blok;
         /*  record from the file  */
   int num,
         /*  record number          */
       pozapl;
         /*  logical mark           */

   long lb_rek,
         /*  number of the file records  */
      poz;
         /*  position in the file        */

   /*   Open the file  */
   /*   Sometimes opening mode   r+
                  instead of   r+b  */
   do {
      printf("Enter the name of the file to be "
                                      "changed: ");
      gets(nazwapliku);
      if ((plik = fopen(nazwapliku, "r+b")) == NULL)
       printf("File doesn't exist - try again!\n");
   }
   while (!plik);

   fseek(plik, 0, 2);
   lb_rek = ftell(plik)/sizeof(blok);

   /*  Loop  - change of the records content  */
   do {
      do {
       printf("\nEnter record nr (0, to finish): ");
       scanf("%d", &num);
       getchar();   /*  jump over \n   */
       pozapl = num < 0 || num > lb_rek ;
      }
      while (pozapl);
      if (num == 0) break;
                  /*  end of changes  */
      poz = (num - 1)*sizeof(blok);
                  /*  current position  */
      fseek(plik, poz, 0);
                  /*  shift and   */
      fread(&blok, sizeof(blok), 1, plik);
                  /*  file read   */
      wyswietl(&blok, num);
                  /*  display and  */
      zmien(&blok);
                  /*  record change  */
      fseek(plik, poz, 0);
                  /*  back in the file  */
      fwrite(&blok, sizeof(blok), 1, plik);
                      /*  record write  */
   }
   while (1);

   /*  End  */
   fclose(plik);
   printf("\n --------------- END --------------- \n");
}

void wyswietl( struct rek *blok, int num)
{
   int i;
   printf("\n\nrecord nr %d \n\n", num);
   printf("Name                      : %s\n",
                               blok->nazw);
   printf("Age                       : %d\n",
                               blok->wiek);
   printf("Number of children        : %d\n",
                                  blok->lb_dz);
   for (i=0; i<blok->lb_dz && i < LBDZ_MAX; i++)
      printf("Age of child nr   %2d : %d\n",
             i+1, blok->wiek_dz[i]);
}

/*  Function changing one record in the file  */
void zmien(struct rek *blok)
{
   char wiersz[127]; /* to read from the keyboard */
   int i;

   printf("\nEnter changes (ENTER,"
          " if you don't change)\n");
   printf("Family name        : ");
   gets(wiersz);
   if (strlen(wiersz)) strcpy(blok->nazw, wiersz);
   printf("age                 : ");
   gets(wiersz);
   if (strlen(wiersz)) blok->wiek=atoi(wiersz);
   printf("Number of children : ");
   gets(wiersz);
   if (strlen(wiersz)) blok->lb_dz=atoi(wiersz);
   for (i=0; i<blok->lb_dz && i<LBDZ_MAX; i++) {
     printf("Age of child number %d : ",i+1);
     gets(wiersz);
     if (strlen(wiersz)) blok->wiek_dz[i]=atoi(wiersz);
   }
}
```