

## EXAMPLES of file operations using the library <stdio.h>

### TEXT files containing sequences of numbers:

- TXT- 1:** Calculator - summing two numbers stored in a text file  
( **fopen**, **fscanf**, **fprintf**, **fclose** )
  
- TXT- 2:** Example (TXT-1) extended with error checking file operations  
(opening the file, reading, writing errors)
  
- TXT- 3:** Summing successive numbers loaded from text file  
(Iterative processing of numbers with loop **while( fscanf(...) == 2 ){...}**  )
  
- TXT- 4:** Iterative processing of sequence of numbers stored in file  
(counting the numbers, calculating the sum, the mean, the max)
  
- TXT- 5:** File operations on ARRAY of numbers  
( **fprintf**, **fscanf**, reading and writing elements of array)
  
- TXT- 6:** Filtering – copying elements matching specified criteria  
( **fscanf**, **fprintf** )
  
- TXT- 7:** Removing the selected elements/numbers from the middle of the file  
(by copying a file into another one, skipping selected values)

## Example TXT-1: Calculators (version without error checking)

Input file: **data.txt**

20.5  
2.5



Output file: **result.txt**

23.00

```
// The program calculates the sum of two floating numbers (of type double).  
// Summed numbers are downloaded from existing disk file "data.txt"  
// Result of the calculation, the sum, is stored in a newly created file "result.txt"  
  
#include <stdio.h>  
void main()  
{  
    double a,b,sum;  
  
    // read the two numbers <double> from the text file into variables <a> and <b>  
    FILE* file;  
    file =fopen("data.txt", "rt");  
    fscanf(file,"%lf %lf",&a,&b);  
    fclose(file);  
  
    // calculate the sum and display the result on console (write to stdout?)  
    sum = a + b;  
    printf("The result: %.2f + %.2f = %.2f\n\n", a, b, sum );  
  
    // write the calculation result (the sum) into the new text file  
    file =fopen("result.txt","wt");  
    fprintf(file,"%0.2f",sum);  
    fclose(file);  
  
    printf("The result has been saved in the file \"result.txt\"");  
    getchar();  
}
```

Used functions:

- fopen** – open the file,  
parameter "**rt**" means „read” „text”  
parameter "**wt**" means „write” „text”
- fscanf** – read formatted data (fields) from a text file,  
parameter "**%lf**" means: convert the number into **double**
- fprintf** – write formatted data to a text file,  
parameter "**%f**" means: print **float** or **double**  
parameter "**.2**" sets precision (2 numbers after decimal dot)
- fclose** – close the file

## Example TXT-2: Calculator (version with "error checking")

```
#include <stdio.h>
void main()
{ double a,b,sum;
  FILE* file = fopen("data.txt", "rt");           ««
  if( file==NULL )                               ««
    { printf("Error while opening the file!");
      getchar();
      return;
    }
  else
    { int read_count;
      read_count = fscanf(file,"%lf %lf",&a,&b);   ««
      fclose(file);
      if( read_count != 2 )                       ««
        { printf("Error while reading the data!");
          getchar();
          return;
        }
      else
        { sum = a + b;
          printf("Result of summation: %.2f + %.2f = %.2f\n\n", a, b, sum);

          file = fopen("result.txt","wt");        ««
          if( file==NULL )                       ««
            { printf("Error while opening the output file!");
              getchar();
              return;
            }
          else
            { int write_count;
              write_count = fprintf(file,"%.2f",sum); ««
              fclose(file);
              if( write_count==EOF )             ««
                { printf("Error saving the result!");
                  getchar();
                  return;
                }
              else
                { printf("The result has been saved in the file");
                  getchar();
                  return;
                }
            }
          }
        }
      }
    }
}
```

### Example TXT-3: Summing successive numbers loaded from text file

*Input file: data.txt*

```
1 1
2 5
5 -4
-500 -50
```

*Output file: sum.txt*

```
2
7
1
-550
```

```
#include <stdio.h>
void main()
{
    int a,b,sum;
    FILE* input_file, *sum_file;

    input_file=fopen("data.txt", "rt");
    sum_file=fopen("sumy.txt", "wt");

    if( input_file==NULL || sum_file==NULL)
    {
        printf("Error while opening files!");
        getchar();
    }
    else
    {

        while( fscanf(input_file,"%d%d",&a,&b)==2 )
        {
            sum = a + b;
            fprintf(sum_file, "%d\n", sum);
            fprintf(stdout, "%d\n", sum);
        }

        fprintf(stdout,"End of program");
        getchar();
    }

    fclose(input_file);
    fclose(sum_file);
}
```

## Przykład TXT-4: Iterative processing of sequence of numbers

Input file: **data.txt**

```
7
-2
1.5
5
```

Result: *numbers on the screen*

```
Count of loaded numbers = 4
Sum = 11.500
Mean = 2.875
```

```
#include <stdio.h>
void main()
{ FILE* file;

file=fopen("data.txt", "rt");
if( file==NULL)
    { printf("Error opening the file!");
      getchar();
    }
else
    { float number, sum=0;
      int count=0;

      while( fscanf(file,"%f",&number)==1 )
      {
          count++;
          sum += number;
          printf("%d – loaded value = %7.2f\n",count,number);
      }
      fclose(file);

      if(count==0)
          printf("No numbers. Can not calculate the mean value");
      else
          { float avg=sum/count;
            printf("\nCount of loaded numbers = %d \n", count);
            printf("Sum = %.3f\n", sum);
            printf("Mean = %.3f\n", avg);
          }
      getchar();
    }
}
```

### Example TXT-5: File operations on ARRAY of numbers

```
#include <stdio.h>
#include <stdlib.h>
void main()
{ const TAB_SIZE_A = 10;
  long tab_A[TAB_SIZE_A];

  for( int i=0; i< TAB_SIZE_A; i++)
    tab_A[i] = rand()%201 -100;

  for(int i=0; i< TAB_SIZE_A; i++)
    printf("tab[%2d]=%4ld\n", i, tab_A[i] );

  FILE* file = fopen("array_of_numers.txt", "wt");
  for( int i=0; i< TAB_SIZE_A; i++)
    fprintf(file,"%4ld\n", tab_A[i] );
  fclose(file);
}
```

```
void main()
{ const int TAB_SIZE_B = 15;
  long tab_B[TAB_SIZE_B]={0};

  printf("\nRead the file and display the loaded array");
  FILE* file = fopen("array_of_numbers.txt", "rt");

  int read_count=0;
  while( read_count< TAB_SIZE_B &&
        fscanf(file,"%ld",&tab_B[ read_count ]) ==1 )
    read_count++;
  fclose(file);

  printf("\nCount of loaded numbera = %d\n",read_count);
  for(int i=0; i<read_count; i++)
    printf("tab[%2d]=%4ld\n", i, tab_B[i] );
  getchar();
}
```

## Example TXT- 6: Filtering elements matching specified criteria

*Input file: numbers.txt*

```
71
4
-12
58
```

*Output file: positive.txt*

```
71
4
58
```

*Output file: negative.txt*

```
-12
```

```
#include <stdio.h>
void main()
{
    FILE* file, *file_positive, *file_negative;

    file = fopen("numbers.txt", "rt");
    file_positive = fopen("positive.txt", "wt");
    file_negative = fopen("negative.txt", "wt");

    if( file && file_positive && file_negative )
    {
        float number;

        while( fscanf(file,"%f",&number)==1 )
        {
            if(number>0)
                fprintf(file_positive,"%f ",number);
            else
                fprintf(file_negative,"%f ",number);
        }
    }

    fclose(file);
    fclose(file_positive);
    fclose(file_negative);
}
```

## Example TXT-7: Removing the numbers from the middle of the file

Input file: numbers.txt

```
-2
4
7
9
15
```

temporary.txt

Output file: numbers.txt

```
-2
4
15
```

```
#include <stdio.h>
void main()
{
    FILE* file = fopen("numbers.txt", "rt");
    FILE* temp_file = fopen("temporary.txt", "wt");

    if( file && temp_file )
    {
        float number;
        while( fscanf(file,"%f",&number)==1 )
            if( number<5 || number>10 )
                fprintf(temp_file,"%f\n",number);
            else
            {
                // Numbers in the range [5,10] are not copied,
                // we can just skip them
                // or display on the screen, to verify the program
                printf("%f\n",number);
            }
    }
    fclose(file);
    fclose(temp_file);

    // physical deletion of initial file "numbers.txt" from the disk
    remove("numbers.txt");

    // renaming the temporary file into "numbers.txt"
    rename("temporary.txt","numbers.txt");
}
```