

PRZYKŁADY OPERACJI PLIKOWYCH z wykorzystaniem biblioteki <stdio.h>

Pliki TEKSTOWE zawierające ciągi liczb:

TXT- 1: Kalkulator – sumowanie dwóch liczb zapisanych w pliku tekstowym
(**fopen, fscanf, fprintf, fclose**)

TXT- 2: Program TXT-1 rozbudowany o kontrolę błędów operacji plikowych
(**błędy: otwarcia pliku, odczytu i zapisu** danych liczbowych)

TXT- 3: Sumator kolejnych par liczb pobieranych z pliku tekstowego
(iteracyjne przetwarzanie ciągu liczb pętlą **while(fscanf(...)==2){...}**)

TXT- 4: Iteracyjne przetwarzanie ciągu liczb wczytywanego z pliku (**fscanf**)
(zliczanie ilości liczb, sumy liczb, wartości średniej)

TXT- 5: Plikowe operacje dla tablicy liczb
(**fprintf, fscanf**, zapisywanie i odczytywanie elementów tablicy)

TXT- 6: Filtrowanie – kopiowanie liczb wybranych w/g zadanego kryterium
(**fscanf, fprintf**)

TXT- 7: Usunięcie wybranych liczb ze środka pliku tekstowego
(poprzez kopiowanie pliku z pominięciem usuwanych wartości)

Pliki BINARNE zawierające ciągi liczb:

BIN-1: Zapis ciągu liczb z klawiatury do pliku binarnego (**fwrite**)

BIN-2: Odczyt i wyświetlenie całej zawartości pliku binarnego (**fread**)

BIN-3: Odczyt wskazanej liczby ze środka pliku binarnego (**fseek, fread**)

BIN-4: Dopisanie trzech liczb (2,4,6) na końcu pliku binarnego (**a - append**)

BIN-5: Zamiana-nadpisanie wartości liczby w pliku binarnym
(**fseek, ftell, fwrite**)

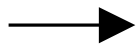
BIN-6: Filtrowanie – kopiowanie liczb w/g zadanego kryterium
za pomocą pętli:

```
while( fread(...)==1 )  
    if( ... )  
        fwrite(...);
```

Przykład TXT-1: Kalkulator (wersja bez kontroli błędów)

Plik wejściowy: **dane.txt**

20.5
2.5



Plik wyjściowy: **wynik.txt**

23.00

```
// Program oblicza sumę dwóch liczb rzeczywistych typu double.  
// Sumowane liczby są pobierane z istniejącego pliku dyskowego "dane.txt"  
// Wynik sumowania jest zapisywany nowo tworzonego pliku "wynik.txt"  
  
#include <stdio.h>  
void main()  
{  
    double a,b,suma;  
  
    // odczyt dwóch liczb <double> z pliku tekstowego do zmiennych <a> i <b>  
    FILE* plik;  
    plik=fopen("dane.txt", "rt");  
    fscanf(plik,"%lf %lf",&a,&b);  
    fclose(plik);  
  
    // wyświetlenie wyniku sumowania na ekranie konsoli (zapis do pliku stdout)  
    suma = a + b;  
    printf("Wynik sumowania: %.2f + %.2f = %.2f\n\n",a,b,suma);  
  
    // zapis wyniku sumowania, tzn. zmiennej <suma>, do pliku tekstowego  
    plik=fopen("wynik.txt","wt");  
    fprintf(plik,"%f",suma);  
    fclose(plik);  
  
    printf("Wynik zostal zapisany w pliku \"wynik.txt\"");  
    getchar();  
}
```

Wykorzystywane funkcje:

- fopen** – otwarcie pliku,
parametry "rt" oznaczają „read” „text”
parametry "wt" oznaczają „write” „text”
- fscanf** – odczyt danych z pliku tekstowego,
parametr "%lf" oznacza odczyt liczby do zmiennej **double**
- fprintf** – zapis danych do pliku tekstowego,
parametr "%f" oznacza zapis liczby **float** lub **double**
parametr ".2" oznacza dokładność 2 miejsc po przecinku
- fclose** – zamknięcie pliku

Przykład TXT-2: Kalkulator (wersja z kontrolowaniem błędów plikowych)

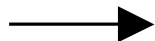
```
#include <stdio.h>
void main()
{ double a,b,suma;
  FILE* plik = fopen("dane.txt", "rt");
  if( plik==NULL )
  { printf("Bład otwarcia pliku danych!");
    getchar();
    return;
  }
  else
  { int ilosc_odczytanych;
    ilosc_odczytanych = fscanf(plik,"%lf %lf",&a,&b);
    fclose(plik);
    if( ilosc_odczytanych != 2 )
    { printf("Bład wczytywania danych!");
      getchar();
      return;
    }
    else
    { suma = a + b;
      printf("Wynik sumowania: %.2f + %.2f = %.2f\n\n",a,b,suma);

      plik = fopen("wynik.txt","wt");
      if( plik==NULL )
      { printf("Bład otwarcia pliku wyników!");
        getchar();
        return;
      }
      else
      { int rezultat_zapisu;
        rezultat_zapisu = fprintf(plik,"%f",suma);
        fclose(plik);
        if( rezultat_zapisu==EOF )
        { printf("Bład zapisywania wyniku!");
          getchar();
          return;
        }
        else
        { printf("Wynik zostal zapisany w pliku \"wynik.txt\");
          getchar();
          return;
        }
      }
    }
  }
}
```

Przykład TXT-3: Sumator par liczb pobieranych z pliku tekstowego

Plik wejściowy: **dane.txt**

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



Plik wyjściowy: **sumy.txt**

```
2
7
1
-550
```

```
// Program oblicza sumy kolejnych par liczb całkowitych typu int.
// Sumowane liczby są pobierane z istniejącego pliku dyskowego "dane.txt"
// Wyniki sumowania są zapisywane w kolejnych wierszach nowego pliku "sumy.txt"

#include <stdio.h>
void main()
{
    int a,b,suma;
    FILE* plik_danych, *plik_sum;

    // otwarcie obu plików: odczytywanego "dane.txt" i zapisywanego "sumy.txt"
    plik_danych=fopen("dane.txt", "rt");
    plik_sum=fopen("sumy.txt", "wt");

    // sprawdzenie poprawności otwarcia plików
    if ( plik_danych==NULL || plik_sum==NULL )
    {
        printf("Bład otwarcia plikow!");
        getchar();
    }
    else
    {

        // iteracyjne przetwarzanie kolejnych par poprawnie odczytanych liczb
        while( fscanf(plik_danych,"%d%d",&a,&b)==2 )
        {
            suma = a + b;
            fprintf(plik_sum,"%d\n",suma);
            fprintf(stdout,"%d\n",suma);
        }

        fprintf(stdout,"Koniec programu",suma);
        getchar();
    }

    fclose(plik_danych);
    fclose(plik_sum);
}
```

Przykład TXT-4: Iteracyjne przetwarzanie ciągu liczb z pliku tekstowego

Plik wejściowy: **dane.txt**

```
7
-2
1.5
5
```

Rezultat: **wydruk na ekranie**

```
Ilosc wczytanych liczb = 4
Suma wczytanych liczb = 11.500
Wartosc srednia wynosi = 2.875
```

```
// Program oblicza: ilość, sumę oraz wartość średnią kolejnych liczb typu float.
// odczytywanych pętlą while( fscanf(...)==1 ) {...}
// z pliku tekstowego "dane.txt"

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

  // otwarcie tekstowego pliku danych do odczytu
  plik=fopen("dane.txt", "rt");
  if( plik==NULL)
  { printf("Bład otwarcia pliku danych!");
    getchar();
  }
  else
  { float liczba,suma=0;
    int licznik=0;

    // pętla iteracyjnego odczytywania i przetwarzania kolejnych liczb float
    while( fscanf(plik,"%f",&liczba)==1 )
    {
      licznik++;
      suma += liczba;
      printf("%d odczytana liczba = %7.2f\n",licznik,liczba);
    }
    fclose(plik);

    // wyświetlenie wyniku przetwarzania (ilości, sumy, średniej) na ekranie
    if(licznik==0)
      printf("Nie wczytano zadnej liczby. Nie mozna policzyc sredniej");
    else
      { float srednia=suma/licznik;
        printf("\nIlosc wczytanych liczb = %d \n", licznik);
        printf("Suma wczytanych liczb = %.3f\n", suma);
        printf("Wartosc srednia wynosi = %.3f\n", srednia);
      }
    getchar();
  }
}
```

Przykład TXT-5: Plikowe operacje dla tablicy liczb <long>

```
// Program tworzy tablicę liczb long, losuje i wyświetla jej zawartość,  
// a następnie zapisuje wszystkie liczby z tej tablicy do nowego pliku "tablica_liczb.txt"  
  
#include <stdio.h>  
#include <stdlib.h>  
void main()  
{ const ROZM_TAB_A = 10;  
  long tab_A[ROZM_TAB_A];  
  
  // losowanie wartości kolejnych elementów tablicy tab_A za pomocą funkcji random  
  for( int i=0; i<ROZM_TAB_A; i++)  
    tab_A[i] = random(201) -100;  
  
  // kontrolne wyświetlenie wartości wszystkich liczb z tablicy tab_A na ekranie  
  for(int i=0; i<ROZM_TAB_A; i++)  
    printf("tab[%2d]=%4ld\n", i, tab_A[i] );  
  
  // zapis wartości elementów tablicy do pliku tekstowego  
  FILE* plik = fopen("tablica_liczb.txt", "wt");  
  for( int i=0; i<ROZM_TAB_A; i++)  
    fprintf(plik, "%4ld\n", tab_A[i] );  
  fclose(plik);  
}
```

```
// Program wczytuje liczby z pliku tekstowego do tablicy tab_B  
// a następnie wyświetla zawartość tablicy na ekranie  
  
void main()  
{ const ROZM_TAB_B = 15;  
  long tab_B[ROZM_TAB_B]={0};  
  
  // otwarcie obu plików: odczytywanego "dane.txt" i zapisywanego "wyniki.txt"  
  printf("\nOdczyt z pliku i wyswietlenie zawartosci tablicy");  
  FILE* plik = fopen("tablica_liczb.txt", "rt");  
  
  // iteracyjny odczyt kolejnych liczb z kontrolowaniem zakresu tablicy (co krótsze?)  
  int ilosc_wczytanych=0;  
  while( ilosc_wczytanych<ROZM_TAB_B &&  
         fscanf(plik,"%ld",&tab_B[ilosc_wczytanych]) ==1 )  
    ilosc_wczytanych++;  
  fclose(plik);  
  
  // kontrolne wyświetlenie zawartości tablicy tab_B po zakończeniu odczytu  
  printf("\nIlosc odczytanych liczb = %d\n",ilosc_wczytanych);  
  for(int i=0; i<ilosc_wczytanych; i++)  
    printf("tab[%2d]=%4ld\n", i, tab_B[i] );  
  getchar();  
}
```

Przykład TXT- 6: Filtrowanie – kopiowanie liczb w/g zadanego kryterium

Plik wejściowy: **liczby.txt**

```
71
4
-12
58
```

Plik wyjściowy: **dodatnie.txt**

```
71
4
58
```

Plik wyjściowy: **ujemne.txt**

```
-12
```

```
// Program rozdziela (kopiuje) liczby odczytywane z pliku "liczby.txt"
// do dwóch nowo utworzonych plików "dodatnie.txt" oraz "ujemne.txt"

#include <stdio.h>
void main()
{
    FILE* plik, *plik_dodatnich, *plik_ujemnych;

    // jednoczesne otwarcie trzech plików: odczytywanego i dwóch wynikowych
    plik=fopen("liczby.txt", "rt");
    plik_dodatnich=fopen("dodatnie.txt", "wt");
    plik_ujemnych=fopen("ujemne.txt", "wt");

    // sprawdzenie poprawności otwarcia plików
    if( plik && plik_dodatnich && plik_ujemnych )
    {
        float liczba;

        // wczytywanie kolejnych liczb z pliku wejściowego "liczby.txt"
        while( fscanf(plik,"%f",&liczba)==1 )
        {
            if(liczba>0)
                fprintf(plik_dodatnich,"%0.8f",liczba);
            else
                fprintf(plik_ujemnych,"%0.8f",liczba);
        }
    }

    // zakończenie operacji kopiowania – zamknięcie plików
    fclose(plik);
    fclose(plik_dodatnich);
    fclose(plik_ujemnych);
}
```

Przykład TXT-7: Usunięcie wybranych liczb ze środka pliku tekstowego

Plik wejściowy: **liczby.txt**

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

pomocniczy.txt

Plik wyjściowy: **liczby.txt**

```
-2
4
15
```

```
// Program usuwa z wnętrza pliku tekstowego liczby nie spełniające zadanego kryterium
// ( w tym przypadku: 5 ≤ usuwana_liczba ≤ 10 )
// poprzez kopiowanie wartości liczb do nowego pliku, z pominięciem usuwanych
```

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

```
void main()
```

```
{ FILE* plik = fopen("liczby.txt", "rt");
```

```
FILE* plik_pomocniczy = fopen("pomocniczy.txt", "wt");
```

```
if( plik && plik_pomocniczy )
```

```
{ float liczba;
```

```
while( fscanf(plik,"%f",&liczba)==1 )
```

```
if( liczba<5 || liczba>10 )
```

```
fprintf(plik_pomocniczy,"%f\n",liczba);
```

```
else
```

```
{
```

```
// liczby z przedziału [5,10] nie są kopiowane,
```

```
// można nic z nimi nie robić (pominąć)
```

```
// lub kontrolnie wydrukować na ekranie konsoli
```

```
printf("%f\n",liczba);
```

```
}
```

```
}
```

```
fclose(plik);
```

```
fclose(plik_pomocniczy);
```

```
// fizyczne usunięcie z dysku początkowego pliku "liczby.txt"
```

```
remove("liczby.txt");
```

```
// zmiana nazwy pomocniczego pliku (zawierającego wybrane liczby) na "liczby.txt"
```

```
rename("pomocniczy.txt","liczby.txt");
```

```
}
```

Usunięcie-skasowanie pliku z dysku: **remove(nazwa_usuwanego_pliku);**

Zmiana nazwy pliku na dysku: **rename(stara_nazwa , nowa_nazwa);**

Przykład BIN-1: Zapis ciągu liczb z klawiatury do pliku binarnego

Plik wejściowy: **stdin**

1	→
2500	→
-18	→
1000000	→

Plik wyjściowy: **liczby.dat** (4 × 8 bajtów)

0x00 0x00 0x00 0x00 0x00 0x00 0x F0 0x 3F
0x00 0x00 0x00 0x00 0x00 0x 88 0x A3 0x 40
0x00 0x00 0x00 0x00 0x00 0x00 0x 32 0x C0
0x00 0x00 0x00 0x00 0x 80 0x 84 0x 2E 0x 41

Zawartość pliku "liczby.dat" oglądanej za pomocą edytora tekstowego (na przykład za pomocą systemowego „Notatnika”) wygląda następująco:

Plik wyjściowy: **liczby.dat** (4 × 8 znaków)

□□□□□□đ?□□□□□□ł@□□□□□□2Ŕ□□□□□€,,A

```
// Program iteracyjnie wczytuje kolejne liczby z klawiatury do zmiennej bufor
// i natychmiast kopiuje zawartość zmiennej bufor (8 bajtów) do pliku binarnego

#include <stdio.h>
void main()
{ FILE* plik = fopen( "liczby.dat", "wb" );
  if(plik)
  {
    // pobranie ilości wczytywanych liczb (w powyższym przykładzie ilość=4 )
    int ilosc_liczb;
    printf("Podaj ile liczb double chcesz zapisać w pliku binarnym: ");
    scanf("%d", &ilosc_liczb);

    double bufor;
    for(int i=0; i<ilosc_liczb; i++)
    {
      printf("\nPodaj kolejną liczbę: ");
      scanf("%lf",&bufor);
      fwrite(&bufor, sizeof(double), 1, plik);
    }
    fclose(plik);
  }
}
```

Zapis binarny: **fwrite(adres_zmiennej, rozmiar_bloku, ilość_bloków, plik)**

Przykład BIN-2: Odczyt i wyświetlenie całej zawartości pliku binarnego

```
// Program iteracyjnie pobiera (wczytuje) z pliku "liczby.dat"
// kolejne 8-bajtowe porcje danych, do zmiennej pomocniczej bufor typu double
// a następnie wyświetla je na ekranie (zapisuje je do pliku tekstowego stdout )

#include <stdio.h>
void main()
{
    FILE* plik = fopen( "liczby.dat", "rb" );
    if( !plik )
    {
        printf("Bład otwarcia pliku");
        getchar();
    }
    else
    {
        double bufor;
        int licznik=0;

        // iteracyjna pętla pobierania kolejnych porcji o rozmiarze sizeof(double)
        while( fread(&bufor, sizeof(double), 1, plik)==1 )
        {

            // wyświetlenie odczytanej liczby na ekranie konsoli
            printf("%f\n", bufor );

            // Tu może nastąpić dalsze przetwarzanie odczytanej liczby
            // np sumowanie: suma=suma+bufor;
            // lub poszukiwanie wartości minimalnej:
            // if( bufor<minimum) minimum=bufor;

            // W tym przykładzie jest tylko samo zliczanie ilości wczytanych liczb:
            licznik++;
        }
        fclose(plik);

        printf("\nKoniec odczytu. Ilość odczytanych liczb = %d",licznik);
    }
}
```

Odczyt binarny: **fread(adres_zmiennej, rozmiar_bloku, ilość_bloków, plik)**

Funkcje **fread** i **fwrite** zwracają ilość poprawnie przetworzonych bloków, tę zwracaną wartość wykorzystuje się do kontrolowania błędów odczytu/zapisu oraz do sterowania zakończeniem pętli odczytującej dane z pliku.

Przykład BIN-3: Odczyt wskazanej liczby ze środka pliku binarnego

```
// Program przesuwa wskaźnik pliku ( "głowicę odczytującą" )
// na wskazane miejsce (odległość w bajtach względem początku pliku)
// a następnie pobiera 8-bajtów do zmiennej liczba typu double
// i wyświetla jej zawartość na ekranie konsoli

#include <stdio.h>
void main()
{ FILE* plik = fopen( "liczby.dat", "rb" );
  if( !plik )
    printf("Bład otwarcia pliku");
  else
    { long pozycja_liczby=30;

      // polecenie przesunięcia wskaźnika pliku
      fseek(plik, pozycja_liczby*sizeof(double), SEEK_SET);

      // odczytanie 8 bajtów ze wskazanego miejsca pliku do zmiennej liczba
      double liczba;
      if( fread(&liczba, sizeof(double), 1, plik)==1 )
        printf("Odczytana liczba = %f\n",liczba);
      else
        printf("Bład odczytu liczby z pliku");
      fclose(plik);
    }
  printf("\nKoniec odczytu. Nacisnij ENTER");
  fflush(stdin);  getchar();
}
```

Przykład BIN-4: Dopisanie trzech liczb (2,4,6) na końcu pliku binarnego

```
#include <stdio.h>
void main()
{
  FILE* plik = fopen( "liczby.dat", "ab" );           ← "a" jak "append"
  if( !plik )
    printf("Bład otwarcia pliku");
  else
    { double liczba;
      for(int i=1; i<=3; i++)
        {
          liczba = i*2;
          fwrite(&liczba, sizeof(double), 1, plik);
        }
      fclose(plik);
    }
}
```

Przykład BIN-5: Zamiana-nadpisanie wartości liczby w pliku binarnym

```
// Program przesuwa wskaźnik pliku na zadaną pozycję względem początku pliku
// a następnie zapisuje tam 8-bajtów ze zmiennej nowa_wartosc typu double

#include <stdio.h>
void main()
{ FILE* plik = fopen( "liczby.dat", "r+b" ); ← "+" bo będzie modyfikacja
  if(plik)
  {
    // odczytanie wielkości pliku (i wyliczenie ilości liczb double)
    fseek(plik,0,SEEK_END);
    long dlugosc_pliku = ftell(plik);
    long ilosc_liczb = dlugosc_pliku / sizeof(double);
    printf("Ten plik zawiera %ld liczb <double>\n\n", ilosc_liczb);

    // ustalenie (wczytanie) pozycji modyfikowanej liczby
    long pozycja_nadpisywanej;
    printf("Podaj pozycję nadpisywanej liczby: ");
    scanf("%ld", &pozycja_nadpisywanej);

    // sprawdzenie czy zadana pozycja jest poprawna ( znajduje się wewnątrz pliku)
    if(pozycja_nadpisywanej>=0 && pozycja_nadpisywanej<ilosc_liczb)
    { double nowa_wartosc;
      printf("\nPodaj nowa wartosc liczby: ");
      scanf("%lf",&nowa_wartosc);

      // przesunięcie wskaźnika pliku i zapis nowej wartości do pliku
      fseek(plik, pozycja_nadpisywanej*sizeof(double), SEEK_SET);
      fwrite(&nowa_wartosc, sizeof(double), 1, plik);
    }
    fclose(plik);
  }
}
```

Przykład BIN-6: Filtrowanie – kopiowanie liczb w/g zadanego kryterium

```
#include <stdio.h> //skopiowanie do drugiego pliku binarnego tylko liczb dodatnich
void main()
{ double liczba;
  FILE* plik = fopen( "liczby.dat", "rb" );
  FILE* plik_dodatnich = fopen("dodatnie.dat", "wb");
  if( plik!=NULL && plik_dodatnich!=NULL )
    while( fread(&liczba, sizeof(double), 1, plik)==1 )
      if(liczba>0)
        fwrite(&liczba, sizeof(double), 1, plik_dodatnich);
  fclose(plik);
  fclose(plik_dodatnich);
}
```