

Operacje na plikach C

Aby otworzyć plik i związać z nim strumień, należy użyć funkcji `fopen()`. Wygląda ona następująco:

```
FILE* fopen(char* nazwa, char* tryb);
```

Funkcja `fopen()` korzysta z pliku nagłówkowego `stdio.h`. Nazwa otwieranego pliku jest wskazywana przez wskaźnik `nazwa`. Musi to być nazwa poprawna w używanym systemie plików. Możliwe wartości trybu znajdują się w poniższym zestawieniu:

Tryb	Opis
r	Otwieranie pliku tekstowego do odczytu
w	Tworzenie pliku tekstowego do zapisu
a	Dołączanie do pliku tekstowego
rb	Otwieranie pliku binarnego do odczytu
wb	Tworzenie pliku binarnego do zapisu
ab	Dołączanie do pliku binarnego
r+	Otwieranie pliku tekstowego do odczytu-zapisu
w+	Tworzenie pliku tekstowego do odczytu-zapisu
a+	Dołączanie do pliku tekstowego do odczytu-zapisu lub jego tworzenie
r+b	Otwieranie pliku binarnego do odczytu-zapisu; można także użyć <code>rb+</code>
w+b	Tworzenie pliku binarnego do odczytu-zapisu; można także użyć <code>wb+</code>
a+b	Dołączanie pliku binarnego do odczytu-zapisu lub jego tworzenie; można także użyć <code>ab+</code>

Przykład 1. Zapis do pliku

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

int main()
{
    FILE* fp; /* używamy metody wysokopoziomowej - musimy mieć zatem
    identyfikator pliku, uwaga na gwiazdkę! */
    char tekst[] = "Hello world";
    if ((fp = fopen("test.txt", "w")) == NULL) {
        printf("Nie mogę otworzyć pliku test.txt do zapisu!\n");
        exit(1);
    }
    fprintf(fp, "%s", tekst); /* zapisz nasz napis w pliku */
    fclose(fp); /* zamknij plik */
} //koniec main
```

Przykład 2.

// Program wyswietlajacy zawartosc pliku

```
#include <stdio.h>

main()
{
    FILE* fp;
    char filename[12];
    int inChar;
    printf("Podaj nazwe pliku, ktory chcesz obejrzec...\n");
    gets(filename);
    if ((fp = fopen(filename, "r")) == NULL)
    {
        printf("\n\n!!! Ten Plik nie istnieje !!!\n");
        exit();
    }
    inChar = getc(fp);
    while (inChar != EOF)
    {
        putchar(inChar);
        inChar = getc(fp);
    }
    fclose(fp);
}
```

Przykład 3.

//Program kopiuje zawartosc jednego pliku do innego (oba wskazane przez uzytkownika)

```
#include <stdio.h>
FILE* inFp;
FILE* outFp;
main()
{
    char inFilename[12];
    char outFilename[12];
    int inChar;
    printf("Podaj nazwe pliku, ktorego kopie chcesz wykonac...\n");
    gets(inFilename);
    printf("Podaj nazwe pliku, do ktorego chcesz kopiowac %s...\n",
inFilename);
    gets(outFilename);
    if ((inFp = fopen(inFilename, "r")) == NULL)
    {
        printf("\n\n!!! Plik %s nie istnieje !!!\n", inFilename);
        exit();
    }
    if ((outFp = fopen(outFilename, "w")) == NULL)
    {
        printf("\n\n!!! Bład otwarcia pliku %s !!!\n", outFilename);
        exit();
    }
    printf("\nKopiowanie...\n");
    while ((inChar = getc(inFp)) != EOF)
    {
        putc(inChar, outFp);
    }
    printf("\nPlik zostal skopiowany...\n");
    fclose(inFp);
}
```

```

        fclose(outFp);
    return 0;
}

```

Przykład 4.

```

//Czytanie pliku liniami
#include<stdio.h>

int main() {
    FILE* fp;
    char buffer[255];

    fp = fopen("alfabet.txt", "r");

    while (fgets(buffer, 255, fp) != NULL) {
        printf("%s\n", buffer);
    }

    fclose(fp);

    return (0);
}

```

Przykład 5.

```

//Czytanie pliku znak po znaku
#include<stdio.h>

int main() {
    FILE* fp;
    char c;

    fp = fopen("alfabet.txt", "r");

    while ((c = fgetc(fp)) != EOF) {
        printf("%c", c);
    }

    fclose(fp);

    return (0);
}

```

Zadania do samodzielnego wykonania:

1. Napisz program, który pobierze z pliku promienie.txt promienie dwóch kół. Twoim zadaniem jest stworzenie tego pliku ręcznie (z poziomu systemu operacyjnego), następnie otwarcie tego pliku za pomocą C oraz wyznaczenie pól kół i zapisanie wyników posortowanych rosnąco (najpierw mniejsze pole, następnie większe) do pliku wynik.txt.
2. Stwórz plik dane.txt z poziomu systemu operacyjnego i wpisz do niego 5 liczb całkowitych. Otwórz ten plik z poziomu C, pobierz wszystkie liczby i do pliku wynik.txt zapisz tylko te, których cyfra jedności kończy się na 0, 3, 8 lub 9.

3. Napisać program zliczający znaki w pliku tekstowym, którego nazwa zczytana jest z wiersza poleceń. Program ma wypisywać w innym pliku zawartość pliku wejściowego i liczbę znaków w pliku.
4. W pliku dane.txt zapisano 1000 numerów PESEL osób urodzonych w latach 1945–2003, po jednym numerze w wierszu. Fragment pliku dane.txt (10 pierwszych wierszy pliku).

```
87062374513
70042949368
57070843873
68100217281
80071113568
58121156393
55042672243
73091424567
89100324122
60062195366
```

Napisz program(y), za pomocą którego(ych) uzyskasz odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku wyniki6.txt, a odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie.

- 4.1. Przedostatnia, dziesiąta cyfra numeru PESEL określa płeć jego właściciela. W numerach PESEL kobiet cyfra ta jest parzysta, u mężczyzn – nieparzysta. Podaj liczbę numerów PESEL kobiet i liczbę numerów PESEL mężczyzn z pliku dane.txt.

Przykład: Dla numerów PESEL zawartych w pliku przyklad.txt odpowiedzią jest 8 numerów dla kobiet i 7 dla mężczyzn.

- 4.2. Cyfry trzecia i czwarta numeru PESEL oznaczają miesiąc urodzenia, przy czym osoby urodzone przed rokiem 2000 mają numer miesiąca podany w sposób naturalny, czyli od 01 do 12, a osoby urodzone w roku 2000 i później mają do numeru miesiąca dodaną liczbę 20, czyli od 21 do 32. Dla danych z pliku dane.txt podaj, ile osób łącznie urodziło się w listopadzie.

Przykład:

Dla numerów PESEL zawartych w pliku przyklad.txt odpowiedzią jest liczba 2.

- 4.3. Poprawność numeru PESEL można obliczyć według podanego poniżej algorytmu:

$$1a_1 + 3a_2 + 7a_3 + 9a_4 + 1a_5 + 3a_6 + 7a_7 + 9a_8 + 1a_9 + 3a_{10} + a_{11}$$

Gdzie a_1 – kolejne cyfry numeru PESEL.

Jeżeli powyższy wynik jest liczbą podzielną przez 10, to numer PESEL jest poprawny, w przeciwnym razie jest błędny. Podaj wszystkie błędne numery PESEL w pliku dane.txt.

