**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Wprowadzenie do Informatyki | **Zadanie** | 8.3 |
| **Autor** | Andrii Godlevskyi | **Grupa** | WCY23KY1S1 |
| **Temat** | Pliki i strumienie | | |

1.Treść zadania

Napisz program, który wczytuje z klawiatury trzy nazwy plików. Dwa pierwsze pliki zawierają posortowane liczby. Program łączy te pliki tak, aby zapisując je do trzeciego pliku w pliku liczby były również posortowane oraz wyświetla na ekranie zawartość trzeciego pliku w wierszach po 10 liczb.

Metoda realizacji

Tworzymy funkcji do: wyświetlania liczb po 10 w wierszu, sortowania liczb. Odczytujemy liczby z plików i wyświetlamy ich na ekranie zanim wywołujemy funkcje sortowania liczb, dalej zapisujemy posortowane liczby do 3 pliku i wyświetlamy ich na ekranie.

1. Założenia / ograniczenia dotyczące danych:

1. Dane wejściowe

nazwa\_pliku we1, nazwa\_pliku we2, nazwa\_pliku wy – wczytywana z klawiatury

1. Dane wyjściowe

Dane z pliku we1, dane z pliku we2, - wyświetlone na ekranie.

Dane z pliku wy - zapisywane do pliku i wyświetlone na ekranie.

1. Realizacja
2. Algorytm

Изображение выглядит как текст, рукописный текст, бумага, чернила

Автоматически созданное описание



1. Kod zródłowy

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_LINE\_LENGTH 100

#define NUMBERS\_PER\_LINE 10

// Funkcja do odczytu liczb z pliku i wyswietlenia ich na ekranie

void readAndDisplayNumbers(const char\* filename) {

FILE\* file = fopen(filename, "r");

if (file == NULL) {

printf("Blad podczas otwierania pliku %s\n", filename);

return;

}

char line[MAX\_LINE\_LENGTH];

int count = 0;

while (fgets(line, MAX\_LINE\_LENGTH, file) != NULL) {

int number;

sscanf(line, "%d", &number);

printf("%d ", number);

count++;

if (count % NUMBERS\_PER\_LINE == 0) {

printf("\n");

}

}

printf("\n");

fclose(file);

}

// Funkcja do sortowania liczb

void sortNumbers(int\* numbers, int count) {

for (int i = 0; i < count - 1; i++) {

for (int j = i + 1; j < count; j++) {

if (numbers[i] > numbers[j]) {

int temp = numbers[i];

numbers[i] = numbers[j];

numbers[j] = temp;

}

}

}

}

// Funkcja do zapisania posortowanych liczb do pliku i wyswietlenia ich na ekranie

void wynik(const char\* filename, int\* numbers, int count) {

FILE\* file = fopen(filename, "w");

if (file == NULL) {

printf("Błąd podczas otwierania pliku %s\n", filename);

return;

}

int printedNumbers = 0;

for (int i = 0; i < count; i++) {

fprintf(file, "%-5d ", numbers[i]);

printf("%-5d ", numbers[i]);

printedNumbers++;

if (printedNumbers % NUMBERS\_PER\_LINE == 0) {

fprintf(file, "\n");

printf("\n");

}

}

printf("\n");

fclose(file);

}

int main() {

char inputFilename[MAX\_LINE\_LENGTH];

char outputFilename[MAX\_LINE\_LENGTH];

//wpisanie danych

printf("Podaj nazwe pliku wejsciowego (\*.txt): ");

scanf("%s", inputFilename);

printf("Podaj nazwę pliku wyjsciowego (\*.txt): ");

scanf("%s", outputFilename);

// Odczytanie liczb z pliku wejsciowego i wyswietlenie ich na ekranie

readAndDisplayNumbers(inputFilename);

// Otwarcie pliku wejściowego do odczytu

FILE\* file = fopen(inputFilename, "r");

if (file == NULL) {

printf("Blad podczas otwierania pliku %s\n", inputFilename);

return 1;

}

// Odczytanie liczb z pliku wejściowego

int numbers[MAX\_LINE\_LENGTH];

int count = 0;

while (fscanf(file, "%d", &numbers[count]) != EOF) {

count++;

}

fclose(file);

// Posortowanie liczb

sortNumbers(numbers, count);

// Zapisanie posortowanych liczb do pliku wyjsciowego i wyświetlenie ich na ekranie

wynik(outputFilename, numbers, count);

return 0;

}

1. Dane wejściowe

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

1. Dane wyjściowe

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

1. Złożoność obliczeniowa algorytmu

O(1)