Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

 Лабараторная работа 18

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Работа з файламі на мове С»

Выканала:

Студэнтка 1 курса 6 группы

Лускіна Вольга Аляксандраўна

Выкладчыца: асс. Андронава М.В.

2023, Мінск

Варыянт 11



Заданне 1

#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fileA; // Робім паказальнік на пачатковы файл

errno\_t errA; // Зменная для памылкі

errA = fopen\_s(&fileA, "fileA.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (errA != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

return 1;

}

int unique\_numbers[100]; // Робім масіў для захоўвання адзінага ўваходу кожнага ліка

int num\_unique = 0; // Стварэнне зменнай для падліку колькасці лікаў, якія не пааўтараюцца

int number; // Стварэнне зменнай для бягучага элемента файла, які чытаем

while (fscanf\_s(fileA, "%d", &number) == 1) { // Цыкл для чытання кожнага элемента файла

int isDuplicate = 0; // Стварэнне кантрольнай зменнай

for (int i = 0; i < num\_unique; i++) { // Цыкл для праверкі ўвахода лікаў, якія не паўтараюцца

if (unique\_numbers[i] == number){

isDuplicate = 1;

break;

}

}

if (!isDuplicate) { // Дадаць лік, які не паўтараецца, у масіў

unique\_numbers[num\_unique] = number;

num\_unique++;

}

}

fclose(fileA);

FILE\* fileB; // Робім паказальнік на файл

errno\_t errB; // Зменная для памылкі

errB = fopen\_s(&fileB, "fileB.txt", "w"); // Адкрыццё файла для запісу

if (errB != 0) {

perror("Памылка адкрыцця файла!");

fclose(fileA);

return 2;

}

for (int i = 0; i < num\_unique; i++) { // Запіс элементаў, якія не паўтараюцца, з аднаго файла ў другі

fprintf(fileB, "%d\t", unique\_numbers[i]);

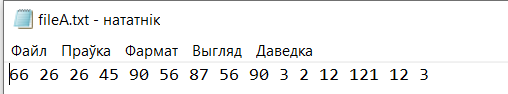
}

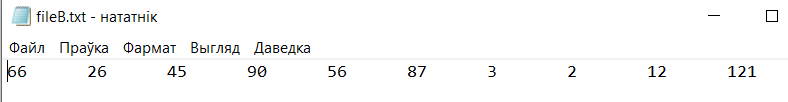
fclose(fileB);

printf("Праграма паспяхова выканана!");

return 0;

}





Заданне 2



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* F1; // Робім паказальнік на пачатковы файл

errno\_t err1; // Зменная для памылкі

err1 = fopen\_s(&F1, "F1.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (err1 != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

return 1;

}

FILE\* F2; // Робім паказальнік на пачатковы файл

errno\_t err2; // Зменная для памылкі

err2 = fopen\_s(&F2, "F2.txt", "w"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (err2 != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

fclose(F1);

return 2;

}

char line[100];

while (fgets(line, sizeof(line), F1)) { // Цыкл для чытання кожнага элемента файла

if (line[strlen(line) - 2] == 'a') {

fputs(line, F2);

}

}

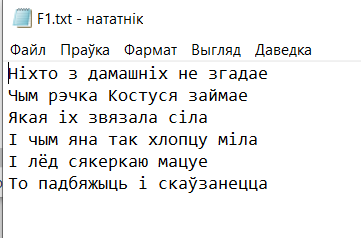
fclose(F1);

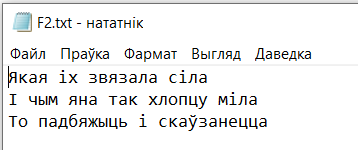
fclose(F2);

printf("Праграма паспяхова выканана!");

return 0;

}





Дадатковыя заданні

Варыянт 13

Заданне 1



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fileA; // Робім паказальнік на пачатковы файл

errno\_t errA; // Зменная для памылкі

errA = fopen\_s(&fileA, "fileA.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (fileA != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

return 1;

}

int numbers[100]; // Робім масіў

int count = 0; // Стварэнне зменнай для падліку колькасці паўтораў

int number; // Стварэнне зменнай для бягучага элемента файла, які чытаем

while (fscanf\_s(fileA, "%d", &number) == 1) { // Цыкл для чытання кожнага элемента файла

numbers[count] = number;

count++;

}

fclose(fileA);

FILE\* fileB; // Робім паказальнік на файл

errno\_t errB; // Зменная для памылкі

errB = fopen\_s(&fileB, "fileB.txt", "w"); // Адкрыццё файла для запісу

if (fileB != 0) {

perror("Памылка стварэння файла!");

fclose(fileA);

return 2;

}

for (int i = 0; i < count; i++) { // Запіс элементаў, якія паўтараюцца больш за два разы, з аднаго файла ў другі

int freq = 0;

int curnum = numbers[i];

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

if (numbers[j] == curnum) {

freq++;

}

}

if (freq > 2) {

bool alreadyWritten = false;

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

if (numbers[k] == curnum) {

alreadyWritten = true;

break;

}

}

if (!alreadyWritten) {

fprintf(fileB, "%d\t", curnum);

}

}

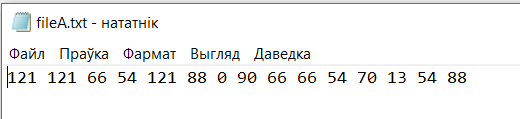
}

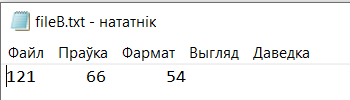
fclose(fileB);

printf("Праграма паспяхова выканана!");

return 0;

}





Заданне 2



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* F1; // Робім паказальнік на пачатковы файл

errno\_t err1; // Зменная для памылкі

err1 = fopen\_s(&F1, "F1.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (err1 != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

return 1;

}

FILE\* F2; // Робім паказальнік на пачатковы файл

errno\_t err2; // Зменная для памылкі

err2 = fopen\_s(&F2, "F2.txt", "w"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (err2 != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

fclose(F1);

return 2;

}

char line[100];

while (fgets(line, sizeof(line), F1)) { // Цыкл для чытання кожнага элемента файла

if (line[0] > '0' && line[0] < '9') {

fputs(line, F2);

}

}

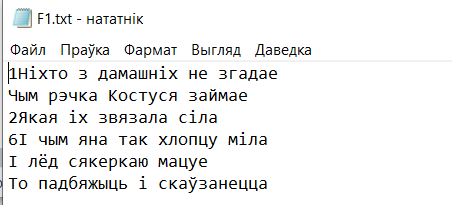
fclose(F1);

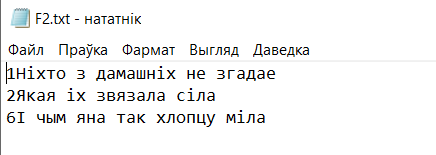
fclose(F2);

printf("Праграма паспяхова выканана!");

return 0;

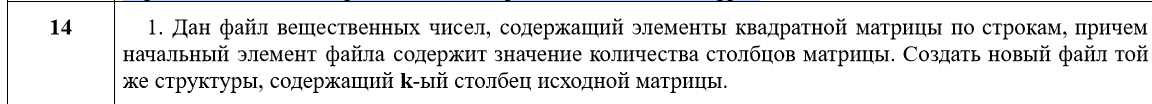
}





Варыянт 4

Заданне 1



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fileA; // Робім паказальнік на пачатковы файл

errno\_t errA; // Зменная для памылкі

errA = fopen\_s(&fileA, "FileA.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (errA != 0) {

perror("Памылка стварэння файла!");

return 1;

}

int amm\_of\_columns; // Робім зменную для падліку колькасці слупкоў з першага элемента

fscanf\_s(fileA, "%d", &amm\_of\_columns); // Адкрыццё пачатковага файла ў рэжыму чытання

double matrix[100][100]; // Робім матрыцу

for (int i = 0; i < amm\_of\_columns; i++) { // Цыкл, каб дадаць элементы файла ў матрыцы

for (int j = 0; j < amm\_of\_columns; j++) {

fscanf\_s(fileA, "%lf", &matrix[i][j]); // Счытваем элементаў з файла

}

}

fclose(fileA);

int k; // Уводзім выбраны нумар слупка з клавіятуры

cout << "Увядзіце нумар слупка, які перанясём у новы файл: ";

cin >> k;

FILE\* fileB; // Робім паказальнік на файл В

errno\_t errB; // Зменная для памылкі

errB = fopen\_s(&fileB, "FileB.txt", "w"); // Адкрыццё файла для запісу

if (errB != 0) {

perror("Памылка стварэння файла!");

fclose(fileA);

return 2;

}

for (int i = 0; i < amm\_of\_columns; i++) { // Цыкл для ўвода патрэбных элементаў матрыцы ў файл

fprintf(fileB, "%.1lf\n", matrix[i][k - 1]);

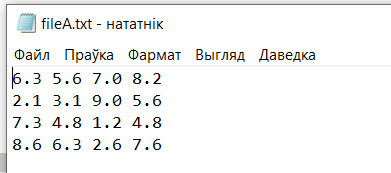
}

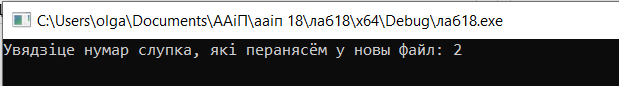
fclose(fileB);

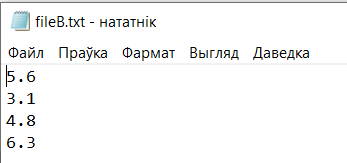
printf("Праграма паспяхова выканана!");

return 0;

}







Заданне 2



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main(){

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* f; // Робім паказальнік на пачатковы файл

errno\_t errF; // Зменная для памылкі

errF = fopen\_s(&f, "f.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (errF != 0) {

perror("Памылка адкрыцця файла!"); // Вывад, калі ёсць памылка

return 1;

}

int unique\_numbers[100]; // Робім масіў

int num\_unique = 0; // Робім зменную для падліку

int number; // Зменная для бягучага элемента

while (fscanf\_s(f, "%d", &number) == 1) { // Цыкл для чытання кожнага элемента файла

int isDuplicate = 0; // Стварэнне кантрольнай зменнай

for (int i = 0; i < num\_unique; i++) { // Цыкл для праверкі ўвахода лікаў, якія не паўтараюцца

if (unique\_numbers[i] == number) {

isDuplicate = 1;

break;

}

}

if (!isDuplicate) { // Дадаем лік, які не паўтараўся ў масіў

unique\_numbers[num\_unique] = number;

num\_unique++;

}

}

fclose(f);

FILE\* g; // Робім паказальнік на новы файл

errno\_t errG; // Зменная для памылкі

errG = fopen\_s(&g, "g.txt", "w"); // Адкрыццё новага файла для запісу

if (errG != 0) {

perror("Памылка адкрыцця файла!");

fclose(f);

return 2;

}

for (int i = 0; i < num\_unique; i++) { // Запіс элементаў, якія не паўтараюцца ў новы файл

fprintf(g, "%d\t", unique\_numbers[i]);

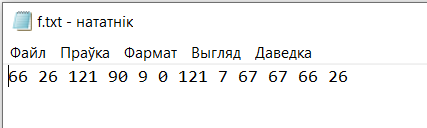
}

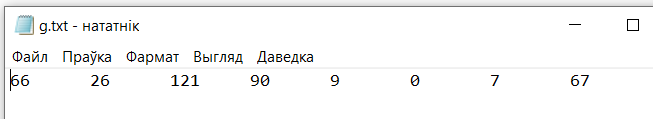
fclose(g);

printf("Праграма паспяхова выканана!");

return 0;

}





Варыянт 4



Заданне 1

#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fileA, \* fileB; // Робім паказальнікi на файлы

int number;

fopen\_s(&fileA, "fileA.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (fileA != 0) {

perror("Памылка адкрыцця файла!");

return 1;

}

fopen\_s(&fileB, "fileB.txt", "w"); // Адкрыццё новага файла для запісу

if (fileB != 0) {

perror("Памылка адкрыцця файла!");

fclose(fileA);

return 1;

}

while (fscanf\_s(fileA, "%d", &number) != EOF) { // Запісваем даданыя лікі

if (number > 0) {

fprintf(fileB, "%d\n", number);

}

}

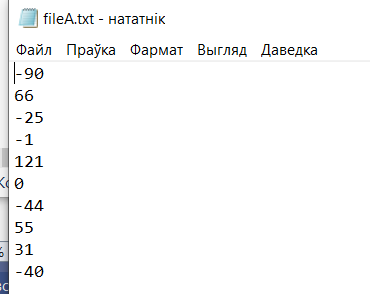
fclose(fileA);

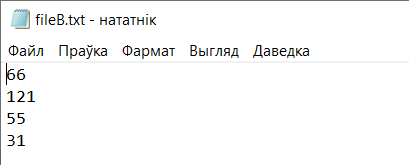
fclose(fileB);

printf("Праграма паспяхова выканана!");

return 0;

}





Заданне 2



#include <iostream>

#include <stdio.h> // Абавязковая бібліятэка

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fileF, \* fileG; // Робім паказальнікi на файлы

int threshold, number;

printf("Увядзіце парогавае значэнне: ");

scanf\_s("%d", &threshold);

fopen\_s(&fileF, "f.txt", "r"); // Адкрыццё пачатковага файла ў рэжыму чытання

if (fileF != 0) {

perror("Памылка адкрыцця файла!");

return 1;

}

fopen\_s(&fileG, "g.txt", "w"); // Адкрыццё новага файла для запісу

if (fileG != 0) {

perror("Памылка адкрыцця файла!");

fclose(fileF);

return 1;

}

while (fscanf\_s(fileF, "%d", &number) != EOF) { // Запісваем лікі, большыя за ўведзеная

if (number > threshold) {

fprintf(fileG, "%d\n", number);

}

}

fclose(fileF);

fclose(fileG);

printf("Праграма паспяхова выканана!");

return 0;

}

