/\*В текстовом файле в одну или несколько строк записаны натуральные числа, разделенные

пробелами. Найти среди них все простые числа,

разделённые пробелами. Найти среди них все простые числа и записать в новый текстовый

файл, рассортировав по возрастанию. Определить простое ли

числа с помощью функции. Использовать динамический массив для хранения чисел.\*/

#pragma warning(disable : 4996) //использование старых функций

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <Windows.h>

using namespace std;

bool IsSimple(int a)//проверка на простоту

{if (a == 1)

return false;

for (int i = 2; i <= a/2; i++)

{

if (a % i == 0)

return false;

}

return true;}

int main()

{ SetConsoleCP(1251);//русские слова

SetConsoleOutputCP(1251);

FILE\* fout = fopen("out1.txt", "wt");//открыть для записи

FILE\* fin = fopen("in1.txt", "rt");

if (fin == NULL)

{ printf("Ошибка открытия входного файла!\n");

fprintf(fout, "%s", "Ошибка открытия входного файла!\n");

fclose(fout);

return 0;}

int kol = 0, val;//val - значения первого найденного числа

if (feof(fin))//проверка на пуcтоту

{ printf("Входной файл не содержит чисел\n");

fprintf(fout, "%s", "Входной файл не содержит чисел\n");

fclose(fout);

fclose(fin);

return 0;}

while (!feof(fin))

{

fscanf(fin, "%d", &val);

if (IsSimple(val))

kol++;

}

fclose(fin);

if (kol == 0)

{

printf("Нет простых чисел\n");

fprintf(fout, "%s", "Нет простых чисел\n");

fclose(fout);

return 0;

}

int\* a = new int[kol];//динамический массив

fopen("in1.txt", "rt");

int k = 0;

while (!feof(fin))

fscanf(fin, "%d", &val);

{

if (IsSimple(val))

a[k++] = val;

fscanf(fin, "%d", &val);

}

int temp;//сортировка пузырьком

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

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

if (a[j] > a[j + 1])

{

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

for (int i = 0; i < kol; i++)

{

fprintf(fout, " %d", a[i]);

}

printf("Готово\n");

fclose(fout);

fclose(fin);

delete[] a;

return 0;}

/\*В текстовом файле в строке записано целое число N (0<N<=10) - размер квадратной

целочисленной матрицы. В следующих строках записаны элементы

строк матрицы, разделенные пробелами. Найти кол-во положительных элементов в верхнем

левом треугольнике матрицы относительно побочной диагонали.

В текстовый файл вывести этот треугольник и найденное кол-во\*/

#pragma warning(disable : 4996)

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <Windows.h>

#include <iostream>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fout = fopen("out2.txt", "wt");

FILE\* fin = fopen("in2.txt", "rt");

if (fin == NULL)

{

printf("Ошибка открытия входного файла!\n");

fprintf(fout, "%s", "Ошибка открытия входного файла!\n");

fclose(fout);

return 0;

}

int a[10][10];

int res = 0, n;

fscanf(fin, "%d", &n);

if (feof(fin))

{

printf("Входной файл не содержит чисел\n");

fprintf(fout, "%s", "Входной файл не содержит чисел\n");

fclose(fout);

fclose(fin);

return 0;

}

if (n < 1 || n>10)

{

printf("Неправильный размер массива\n");

fprintf(fout, "%s", "Неправильный размер массива\n");

fclose(fout);

fclose(fin);

return 0;

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

fscanf(fin, "%d", &a[i][j]);

if (feof(fin))

{

printf("Недостаточно чисел\n");

fprintf(fout, "%s", "Недостаточно чисел\n");

fclose(fout);

fclose(fin);

return 0;

}

}

}

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n - i; j++)

{

if (a[i][j] > 0)

res++;

fprintf(fout, "%5d", a[i][j]);

}

fprintf(fout, " %s\n", " ");

}

printf("Зайдите в файл\n");

fprintf(fout, " %s\n", " Количество положительных элементов в данном треугольнике:");

fprintf(fout, " %d\n", res);

fclose(fout);

fclose(fin);

return 0;}

/\*В первой строке текстового файла записаны подряд символы-разделители между словами.

В последующих строках записаны слова, разделенные ими.

Выбрать из текста слова, являющиеся вещественными числами и записать их в

первую строку выходного файла. Размер строк не превосходит 255 символов.

Отсортировать по возрастанию модуля и записать во второй строке. \*/

#pragma warning(disable : 4996)

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <Windows.h>

#include <iostream>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fout = fopen("out3.txt", "wt");

FILE\* fin = fopen("in3.txt", "rt");

if (fin == NULL)

{

printf("Ошибка открытия входного файла!\n");

fprintf(fout, "%s", "Ошибка открытия входного файла!\n");

fclose(fout);

return 0;

}

char d[257], s1[257];

if (fgets(d, 256, fin) == NULL)//если ничего не прочитали - пустой

{

printf("Входной файл пустой\n");

fprintf(fout, "%s", "Входной файл пустой\n");

fclose(fout);

fclose(fin);

return 0;

}

d[strlen(d) - 1] = 0;//строка с разделителями

if (d[0] == 0)

{

printf("Во входном файле нет строки 1 с разделителями\n");

fprintf(fout, "%s", "Во входном файле нет строки 1 с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

if (fgets(s1, 256, fin) == NULL)

{

printf("Входной файл содержит только 1 строку с разделителями\n");

fprintf(fout, "%s", "Входной файл содержит только 1 строку с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

char\* p;

double a[1000], dd;//массив и длина

int n = 0;

char\* end\_ptr;//указатель на конец того, что ищем (вещественные числа)

do

{

s1[strlen(s1)] = 0;

if ((strlen(s1)) == 0)

continue;

p = strtok(s1, d);

while (p != 0)

{

dd = strtod(p, &end\_ptr);

if (\*end\_ptr == '\0')

a[n++] = dd;

p = strtok(NULL, d);

}

} while (fgets(s1, 256, fin) != NULL);

if (n == 0)

{

printf("Входной файл не содержит чисел\n");

fprintf(fout, "%s", "Входной файл не содержит чисел\n");

fclose(fout);

fclose(fin);

return 0;

}

else

{

for (int i = 0; i < n; i++)//сортировка пузырьком по модулю

{

fprintf(fout, " %g", a[i]);

}

fprintf(fout, " \n");

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n - i - 1; j++)

{

if (fabs(a[j]) > fabs(a[j + 1]))

{

double temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

for (int i = 0; i < n; i++)

{

fprintf(fout, " %g", a[i]);

}

printf("Задача выполнена успешно\n");

}

fclose(fout);

fclose(fin);

return 0;

}

/\*В первой строке - символы-разделители слов в тексте, который записан в

последующих строках. Выбрать из текста слова, которые являются целыми

числами и записать их в первую строку выходного файла. Для каждого числа

найти сумму его цифр и записать эти суммы во вторую строку выходного файла.

Найти число с максимальной суммой составляющих его цифр и записать в третью строку

в виде (число \_пробел\_ сумма цифр). Если таких чисел несколько - то

каждое в новой строке.\*/

#pragma warning(disable : 4996)

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

int summa(int chislo)//сумма цифр числа

{

int k = 0;

while (chislo != 0)

{

k = k + (chislo % 10);

chislo = chislo / 10;

}

return k;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fout = fopen("out4.txt", "wt");

FILE\* fin = fopen("in4.txt", "rt");

if (fin == NULL)

{

printf("Ошибка открытия входного файла!\n");

fprintf(fout, "%s", "Ошибка открытия входного файла!\n");

fclose(fout);

return 0;

}

char d[257], s1[257];

if (fgets(d, 256, fin) == NULL)

{

printf("Входной файл пустой\n");

fprintf(fout, "%s", "Входной файл пустой\n");

fclose(fout);

fclose(fin);

return 0;

}

d[strlen(d) - 1] = 0;

if (d[0] == 0)

{

printf("Во входном файле нет строки с разделителями\n");

fprintf(fout, "%s", "Во входном файле нет строки с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

if (fgets(s1, 256, fin) == NULL)

{

printf("Входной файл содержит только 1 строку с разделителями\n");

fprintf(fout, "%s", "Входной файл содержит только 1 строку с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

char\* p;

int a[1000], ii;

int n = 0;

char\* end\_ptr;

do

{

s1[strlen(s1) - 1] = 0;

if ((strlen(s1)) == 0)

continue;

p = strtok(s1, d);

while (p != 0)

{

ii = strtol(p, &end\_ptr, 10);

if (\*end\_ptr == '\0')

a[n++] = ii;

p = strtok(NULL, d);

}

} while (fgets(s1, 256, fin) != NULL);

if (n == 0)

{

printf("Входной файл не содержит чисел\n");

fprintf(fout, "%s", "Входной файл не содержит чисел\n");

fclose(fout);

fclose(fin);

return 0;

}

else

{

for (int i = 0; i < n; i++)

{

fprintf(fout, " %d", a[i]);

}

fprintf(fout, " \n");

int arr[1000], max = 0;

for (int i = 0; i < n; i++)

{

fprintf(fout, " %d", summa(abs(a[i])));

}

for (int i = 0; i < n; i++)

{

arr[i] = summa(abs(a[i]));

if (arr[i] >= max)

{

max = arr[i];

}

}

for (int i = 0; i < n; i++)

{

arr[i] = summa(abs(a[i]));

if (arr[i] == max)

{

fprintf(fout, " \n");

fprintf(fout, " %d", a[i]);

fprintf(fout, " %d", arr[i]);

}

}

printf("Задача выполнена успешно\n");

}

fclose(fout);

fclose(fin);

return 0;

}

/\*В первой - разделители. Каждую следующую строку преобразовать по правилу:

удалить из нее слова, состоящие только из цифр. Преобразованные строки

записать в выходной файл.\*/

#pragma warning(disable : 4996)

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <Windows.h>

#include <iostream>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

FILE\* fout = fopen("out5.txt", "wt");

FILE\* fin = fopen("in5.txt", "rt");

if (fin == NULL)

{

printf("Ошибка открытия входного файла!\n");

fclose(fout);

return 0;

}

char d[257], s1[256];

string numb = "0123456789";

if (fgets(d, 256, fin) == NULL)

{

printf("Входной файл пустой\n");

fclose(fout);

fclose(fin);

return 0;

}

d[strlen(d) - 1] = 0;

if (d[0] == 0)

{

printf("Во входном файле нет строки с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

if (fgets(s1, 256, fin) == NULL)

{

printf("Входной файл содержит только строку с разделителями\n");

fclose(fout);

fclose(fin);

return 0;

}

char\* p;

double a[1000], dd;

int n = 0;

char\* end\_ptr;

do

{

s1[strlen(s1) - 1] = 0;

if ((strlen(s1)) == 0)

continue;

string str = s1;

string s;

unsigned pos1 = 0, pos2;

while ((pos1 = str.find\_first\_not\_of(d, pos1)) != string::npos)

{

pos2 = str.find\_first\_of(d, pos1);

s = str.substr(pos1, pos2 - pos1);

if (s.find\_first\_not\_of(numb, 0) == string::npos)

{

str.replace(pos1, pos2 - pos1, "");

}

pos1 = pos2;

}

fprintf(fout, str.c\_str());

fprintf(fout, "\n");

} while (fgets(s1, 256, fin) != NULL);

printf("задача выполнена успешно\n");

fclose(fout);

fclose(fin);

return 0;

}

//В текстовом файле записаны строки. Строки содержит два слова, разделенные одним

или несколькими пробелами. Проверить, можно ли из букв первого слова составить второе,

используя любую букву первого слова не более одного раза.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <string.h>

using namespace std;

int main()

{

FILE\* in;

if ((in = fopen("input.txt", "r")) == NULL)

{

cout << "File could not be opened" << endl;

system("pause");

return -1;

}

FILE\* out;

if ((out = fopen("output.txt", "w")) == NULL)

{

cout << "File could not be opened1" << endl;

system("pause");

return -1;

}

const int nmax = 30;

char\* word1 = new char[30];

fscanf(in, "%s", word1);

if (feof(in))

{

cout << "File is empty" << endl;

system("pause");

return -1;

}

char\* word2 = new char[30];

fscanf(in, "%s", word2);

char\* cpyword1 = new char[30];

bool flag = true;

while (feof(in) == NULL)

{

strcpy(cpyword1, word1);

int dlina1 = strlen(word1);

int dlina2 = strlen(word2);

int a = 0;

if (dlina1 >= dlina2)

{

for (int i = 0; i < dlina2; i++)

for (int j = a; j < dlina1; j++)

{

if (word2[i] == word1[j])

{

int tmp;

a++;

break;

}

}

}

if (a == dlina2)

{

fprintf(out, "%s %s - yes\n", cpyword1, word2);

}

else

{

fprintf(out, "%s %s - no\n", cpyword1, word2);

}

fscanf(in, "%s", word1);

fscanf(in, "%s", word2);

}

system("pause");

return 0;

}

//разбиение файла

//штуки, чтобы не капризнячала вижла

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_DEPRECATE

#define \_CRT\_NONSTDC\_NO\_DEPRECATE

#pragma warning(disable : 4996)

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

FILE\* in, \* out1, \* out2;

setlocale(LC\_ALL, ".1251");

in = fopen("input.txt", "rt");

if (!in)

{

cout << "Входной файл не открывается" << endl;

system("pause");

return -1;

}

char\* s = new char[256];

fgets(s, 255, in);

if (feof(in))

{

cout << "Входной файл пустой" << endl;

fclose(in);

system("pause");

return -1;

}

out1 = fopen("output1.txt", "wt");

if (!out1)

{

cout << "Выходной файл1 не открывается" << endl;

fclose(in);

system("pause");

return -1;

}

out2 = fopen("output2.txt", "wt");

if (!out2)

{

cout << "Выходной файл2 не открывается" << endl;

fclose(in);

fclose(out1);

system("pause");

return -1;

}

char dividers[] = "\n ";

char\* s1 = new char[256];

char\* s2 = new char[256];

char\* memory = new char[256];

int counter\_for\_checking\_lines\_in\_file1 = 0;

int counter\_for\_checking\_lines\_in\_file2 = 0;

while (!feof(in))

{

char\* word;

strcpy(memory, s);

word = strtok(s, dividers);

int counter\_for\_checking\_items = 0;

while (word)

{

for (int i = 0; i < strlen(word); i++)

{

if ((isalpha(word[i]) == NULL) && (isdigit(word[i])

== NULL))

counter\_for\_checking\_items++;

if (counter\_for\_checking\_items != 0)

{

if (counter\_for\_checking\_lines\_in\_file2 == 0)

{

strcpy(s2, memory);

counter\_for\_checking\_lines\_in\_file2++;

break;

}

else

{

strcat(s2, memory);

break;

}

}

}

if (counter\_for\_checking\_items != 0)

break;

word = strtok(NULL, dividers);

}

if (counter\_for\_checking\_items == 0)

{

if (counter\_for\_checking\_lines\_in\_file2 == 0)

{

counter\_for\_checking\_lines\_in\_file1++;

strcpy(s1, memory);

}

else

strcat(s1, memory);

}

fgets(s, 255, in);

}

fputs(s1, out1);

fputs(s2, out2);

fclose(in);

fclose(out1);

fclose(out2);

delete[] s;

delete[] s1;

delete[] s2;

delete[] memory;

system("pause");

return 0;

}

//Палиндромы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

bool isPalindrom (string word){

int flag = 0;

for(int i = 0; i < word.length() / 2; i++){

if(word[i] == word[word.length() - i - 1])

flag++;

}

if(flag == word.length() / 2)

return true;

return false;

}

int main(){

ifstream fin("input.txt");

ofstream fout("output.txt");

if (!fin){

cout << "Input file is not opened!" << endl;

}

if (!fout){

cout << "Input file is not opened!" << endl;

}

if (fin.peek() == EOF){

cout << "Input file is empty!" << endl;

fin.close();

fout.close();

}

string str, str2, delim;

string \*word = new string[256];

getline(fin, delim);

int pos1 = 0, pos2 = 0;

int j = 0;

while (!fin.eof()){

getline (fin, str);

int n = 0;

while ((pos1 = (int)str.find\_first\_not\_of(delim, pos1)) != -1){

pos2 = (int)str.find\_first\_of(delim, pos1);

str2 = str.substr(pos1, pos2 - pos1);

word[n++] = str2;

pos1 = pos2;

}

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

if (isPalindrom(word[i])){

fout << word[i] << ' ';

j++;

}

}

pos1 = 0;

}

if (j == 0){

fout << "No words palindroms!" << endl;

}

cout << "Look in your output file" << endl;

delete[] word;

fin.close();

fout.close();

}

/\*5. Составить программу решения следующей задачи:

В текстовом файле записаны строки длиной не более 255 символов, слова в которых разделены хотя бы одним

пробелом или знаками препинания. В каждой строке найти слова, являющиеся простыми числами и поменять местами

максимальное и минимальное. Если максимальных и/или минимальных чисел несколько, то брать последнее. Результаты

записать в новый текстовый файл, сохраняя все разделители. \*/

#include <iostream>

#include <cmath>

#include <fstream>

#include <string>

using namespace std;

ifstream fin;

ofstream fout;

bool simple(int k)

{

int g = 0;

for (int i = 1; i < k / 2 + 1; i++)

{

if (k % i == 0)

g++;

if (g > 1)

return false;

}

if (g <= 1)

return true;

}

int main()

{

setlocale(LC\_ALL, "rus");

fin.open("input.txt");

fout.open("output.txt");

if (!fin.is\_open())

{

cout << "Входной файл не открылся!\n";

fin.close();

fout.close();

system("pause");

return 0;

}

if (!fout.is\_open())

{

cout << "Выходной файл не открылся\n";

fin.close();

fout.close();

system("pause");

return 0;

}

string str;

getline(fin, str);

if (str == "")

{

cout << "Входной файл пустой\n";

fin.close();

fout.close();

system("pause");

return 0;

}

do

{

int a = 0, b;

int kolvo = 0;

int fmax = 0, fmin = 0, f1, f2;

cout << "str = " << str << endl;

while ((a = str.find\_first\_not\_of(" ,.;:", a)) != -1)

{

b = str.find\_first\_of(" ,.;:", a);

string d = str.substr(a, b - a);

int f;

if (d.find\_first\_not\_of("1234567890+", 0) != -1)

f = -1;

else

f = atoi(d.c\_str());

cout << "d = " << d << "\tf = " << f << endl;

if (f > 0 && simple(f))

{

if (kolvo == 0)

{

fmax = fmin = f;

f1 = f2 = a;

}

else

{

if (f >= fmax)

{

fmax = f; f1 = a;

}

if (f <= fmin)

{

fmin = f; f2 = a;

}

}

kolvo++;

}

a = b;

}

if (fmax == fmin || kolvo <= 1)

{

fout << str << endl;

}

else

{

cout << "fmin = " << fmin << "\tfmax = " << fmax << "\tf1 = " << f1 << "\tf2 = " << f2 << endl;

if (f1 < f2)

{

str.replace(f2, to\_string(fmin).length(), to\_string(fmax));

str.replace(f1, to\_string(fmax).length(), to\_string(fmin));

}

else

{

str.replace(f1, to\_string(fmax).length(), to\_string(fmin));

str.replace(f2, to\_string(fmin).length(), to\_string(fmax));

}

fout << str << endl;

}

} while (getline(fin, str));

fin.close();

fout.close();

system("pause");

return 0;

}