**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

Індивідуальне завдання № 3  
з дисципліни «Алгоритмізація та програмування»

Тема роботи: «Обробка рядків символів.»

**Варіант** **№ 3**

Виконав(-ла) студент(-ка)

групи АнД-11

Яковкін Микола Андрійович

Перевірив(-ла):

Київ 2021

**Завдання 1**

**Видалити з S перше входження буквосполучення «are»;**

**Реалізація на мові програмування С**

* 1. **Тестування програми**

1. Для перевірки вставимо рядок “Hello how are you?”. Програма має повернути рядок “Hello how you?”.

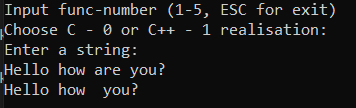


Рисунок 1.1 – Тестування програми.

1. Для перевірки вставимо рядок “are you on the area?”. Програма має повернути рядок “ you on the area?”.

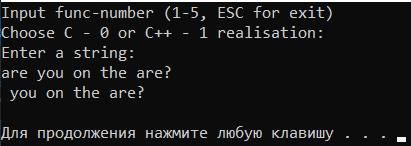
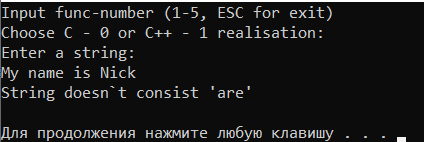


Рисунок 1.2 – Тестування програми.

1. Для перевірки вставимо рядок “My name is Nick ”. Програма має повернути повідомлення про те що рядок не має буквосполучення «are».

  
Рисунок 1.3 – Тестування програми.

**1.2 Текст програмною мовою С.**

**void** taskOneC()

{

**char** str1[MAX];

**char** str2[MIN] = "are";

**puts**("Enter a string: ");

**fgets**(str1, MAX, **stdin**);

**char**\* p = **strstr**(str1, str2);

**if** (p)

{

**strcpy**(p, strlen(str2) + p);

**printf**(str1);

}

else

{

**printf**("String doesn`t consist 'are' \n");

}

}

**Реалізація на мові програмування С++**

**1.3 Тестування програми**

1. Для перевірки вставимо рядок “Where are you from? ”. Програма має повернути рядок “ Where you from?”.

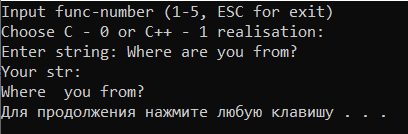


Рисунок 1.4 – Тестування програми.

1. Для перевірки вставимо рядок “awareness”. Програма має повернути рядок “ awness”.

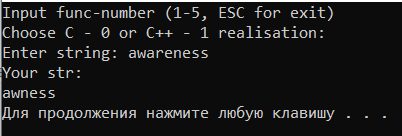
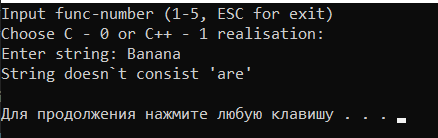


Рисунок 1.5 – Тестування програми.

1. Для перевірки вставимо рядок “Banana”. Програма має повернути повідомлення про те що рядок не має буквосполучення «are».

  
Рисунок 1.6 – Тестування програми.

**1.4 Текст програмною мовою С++.**

**void** taskOneCPP()

{

**string** str1;

**string** str2 = "are";

cout << "Enter string: ";

**getline**(**cin**, str1);

**int** location = str1.**find**(str2);

**int** size = str2.**length**();

**if** (location != std::string::npos)

{

str1.**erase**(location, size);

cout << "Your str: " << **endl** << str1;

}

**else**

cout << "String doesn`t consist 'are'" << endl;

}

**Завдання 2**

**Після першої букви «a» вставити буквосполучення «bc»**

**Реалізація на мові програмування С**

**2.1 Тестування програми**

1. Для перевірки вставимо рядок “Banana ”. Програма має повернути рядок “ Babcnana ”.

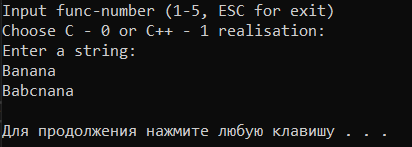


Рисунок 2.1 – Тестування програми.

1. Для перевірки вставимо рядок “ brunfelsia” Програма має повернути рядок “ brunfelsiabc”

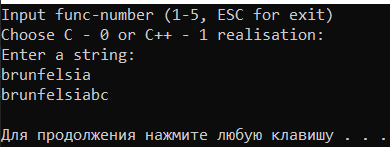
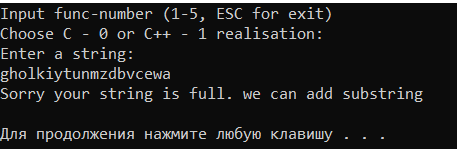


Рисунок 2.2 – Тестування програми.

1. Для перевірки вставимо рядок “gholkiytunmzdbvcewa ”. Програма має повернути повідомлення про те що рядок не має більше місця.

  
Рисунок 2.3 – Тестування програми.

**2.2 Текст програмною мовою С.**

**void** taskTwoC()

{

**char** letter = 'a';

**char** str1[20];

**char** str2[MIN] = "bc";

**char** temp[MAX] = "\0";

**puts**("Enter a string: ");

**fgets**(str1, 20, **stdin**);

if ((**strlen**(str1) + **strlen**(str2) < 19))

{

if (**strchr**(str1, letter))

{

**char**\* p = strchr(str1, letter) + 1;

**strcpy**(temp, p);

**strcpy**(p, str2);

p += **strlen**(str2);

**strcpy**(p, temp);

printf(str1);

}

**else**

**printf**("Sorry your string doesn`t consist letter 'a'\n");

}

**else**

**printf**("Sorry your string is full. we can add substring\n");

}

**Реалізація на мові програмування С++**

**2.3 Тестування програми**

1. Для перевірки вставимо рядок “man ”. Програма має повернути рядок “ mabcn”.

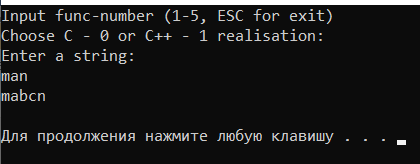


Рисунок 2.4 – Тестування програми.

1. Для перевірки вставимо рядок “awareness”. Програма має повернути рядок “ abcwareness ”.

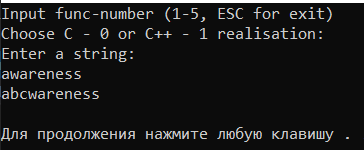
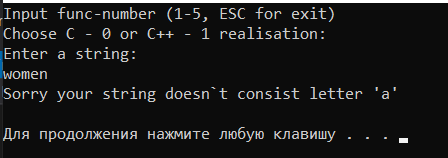


Рисунок 2.5 – Тестування програми.

1. Для перевірки вставимо рядок “women”. Програма має повернути повідомлення про те що рядок не має літери «a».

  
Рисунок 2.6 – Тестування програми.

**2.4 Текст програмною мовою С++.**

**void** taskTwoCPP()

{

**string** str1;

**string** str2 = "bc";

**char** letter = 'a';

cout << "Enter string: ";

**getline**(**cin**, str1);

**int** location = str1.**find**(letter) + 1;

**if** (location)

{

str1.**insert**(location, str2);

cout << "Your str: " << **endl** << str1;

}

**else**

{

cout << "Sorry your string doesn`t consist letter 'a'" << **endl**;

}

}

**Завдання 3**

**В рядок X скопіювати 6 символи з рядка S, починаючи з позиції**

**наступної за першою буквою «a»**

**Реалізація на мові програмування С**

* 1. **Тестування програми**

1. Для перевірки вставимо рядок “Banana ”. Програма має повернути рядок “ nana ”.

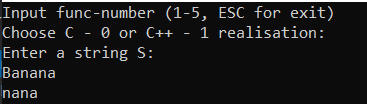


Рисунок 3.1 – Тестування програми.

1. Для перевірки вставимо рядок “ brunfelsia” Програма має повернути рядок “ ”

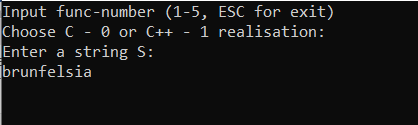
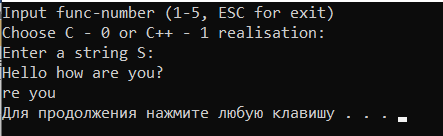


Рисунок 3.2 – Тестування програми.

1. Для перевірки вставимо рядок “Hello how are you?” Програма має повернути рядок “re you”

  
Рисунок 3.3 – Тестування програми.

**3.2 Текст програмною мовою С.**

**void** taskThreeC()

{

**char** S[MAX];

**char** X[7] = "\0";

**char** letter = 'a';

**puts**("Enter a string S: ");

**fgets**(S, MAX, stdin);

**char**\* p = strchr(S, letter) + 1;

**strncat**(X, p, 6);

**printf**(X);

}

**Реалізація на мові програмування С++**

**3.3 Тестування програми**

1. Для перевірки вставимо рядок “man”. Програма має повернути рядок “n”.

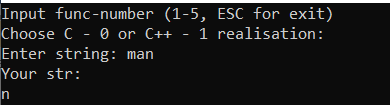


Рисунок 3.4 – Тестування програми.

2. Для перевірки вставимо рядок “awareness”. Програма має повернути рядок “ warene”.

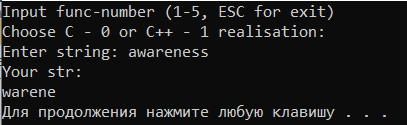
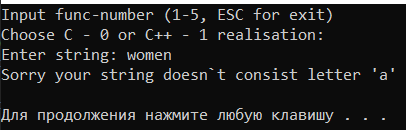


Рисунок 3.5 – Тестування програми.

3.Для перевірки вставимо рядок “women”. Програма має повернути повідомлення про те що рядок не має літери «a».

  
Рисунок 3.6 – Тестування програми.

**3.4 Текст програмною мовою С++.**

**void** taskThreeCPP()

{

**string** S;

**string** X;

**char** letter = 'a';

cout << "Enter string: ";

**getline**(cin, S);

**int** location = S.find(letter) + 1;

if (location)

{

X.**append**(S, location, 6);

cout << "Your str: " << **endl** << X;

}

**else**

cout << "Sorry your string doesn`t consist letter 'a'" << **endl**;

}

**Завдання 4**

**Видалити з S всі входження «\*\*\*».**

**Реалізація на мові програмування С**

**4.1 Тестування програми**

Для перевірки вставимо рядок “\*\*\*N\*\*i\*\*\*c\*\*\*k ”. Програма має повернути рядок “ Nick ”.

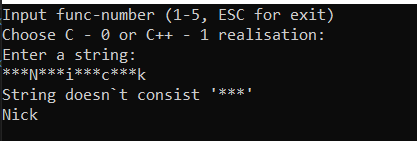


Рисунок 4.1 – Тестування програми.

Для перевірки вставимо рядок “ \*\*i\*\*k” Програма має повернути рядок “\*\*i\*\*k”

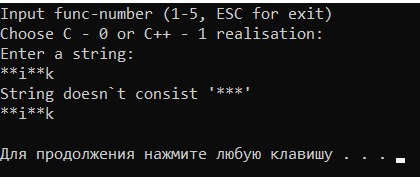
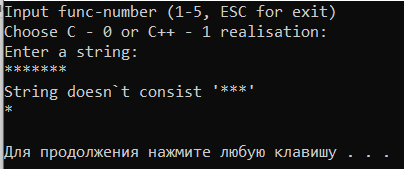


Рисунок 4.2 – Тестування програми.

Для перевірки вставимо рядок “\*\*\*\*\*\*\*?” Програма має повернути рядок “\*”

  
Рисунок 3.3 – Тестування програми.

**4.2 Текст програмною мовою С.**

**void** taskFourC()

{

**char** str1[MAX];

**char** str2[MIN] = "\*\*\*";

**puts**("Enter a string: ");

**fgets**(**str1**, MAX, stdin);

**while** (**true**)

{

**char**\* p = strstr(str1, str2);

**if** (p)

{

**strcpy**(p, **strlen**(str2) + p);

//printf(str1);

}

**else**

{

**printf**("String doesn`t consist '\*\*\*' \n");

**printf**(str1);

**break**;

}

}

}

**Реалізація на мові програмування С++**

**4.3 Тестування програми**

1. Для перевірки вставимо рядок “ Hello \*\*\*my \*\*\*name \*\*\*is Nick”. Програма має повернути рядок “Hello my name is Nick ”.

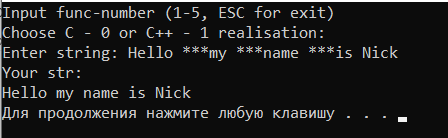


Рисунок 4.4 – Тестування програми.

2. Для перевірки вставимо рядок “\*\*\*\*\*\*\*\*\*”. Програма має повернути рядок “”.

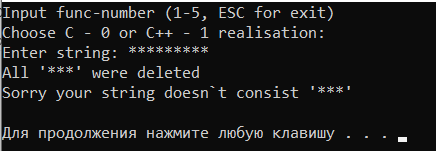
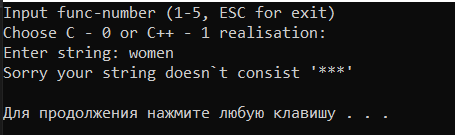


Рисунок 4.5 – Тестування програми.

3.Для перевірки вставимо рядок “women”. Програма має повернути повідомлення про те що рядок не має «\*\*\*».

  
Рисунок 4.6 – Тестування програми.

**4.4 Текст програмною мовою С++.**

**void** taskFourCPP()

{

**int** n = 0;

**string** str1;

**string** str2 = "\*\*\*";

cout << "Enter string: ";

**getline**(**cin**, str1);

**while** (str1.**find**(str2) < MAX)

{

**int** location = str1.**find**(str2);

**if** (location)

{

**int** size = str2.**length**();

str1.**erase**(location, size);

n = 1;

}

**else**

{

cout << "All '\*\*\*' were deleted" << **endl**;

**break**;

}

}

**if**(n)

cout << "Your str: " << **endl** << str1;

**else**

cout << "Sorry your string doesn`t consist '\*\*\*'" << **endl**;

}

**Завдання 5**

**Знайти найбільше ціле та дійсне число в реченні. Впорядкувати**

**декілька речень за зростанням цілих та за спаданням дійсних чисел,**

**що вони містять. Результат подати у відповідних масивах покажчиків**

**на його елементи.**

**Реалізація на мові програмування С**

**5.1 Тестування програми**

1. Для перевірки вставимо у файл:

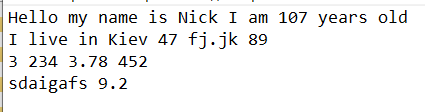


Рисунок 5.1 – Вхідні данні.

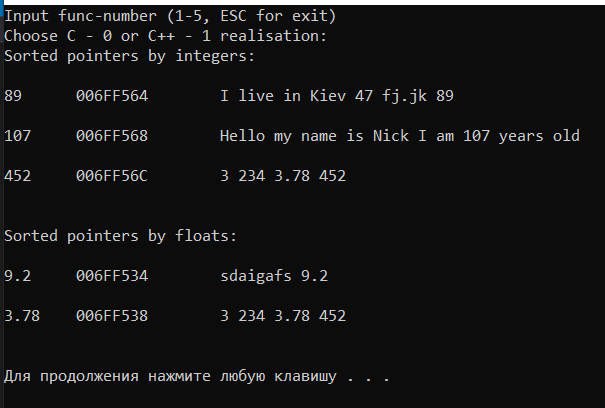


Рисунок 5.2 – Тестування програми.

1. Для перевірки вставимо у файл:

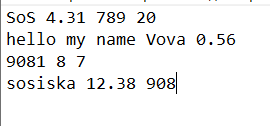


Рисунок 5.3 – Вхідні данні.

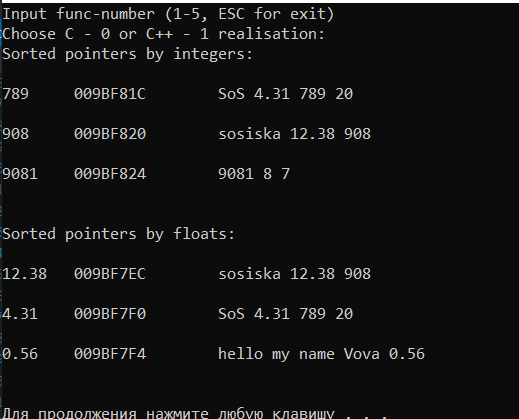


Рисунок 5.4 – Тестування програми.

1. Для перевірки вставимо у файл:

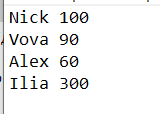
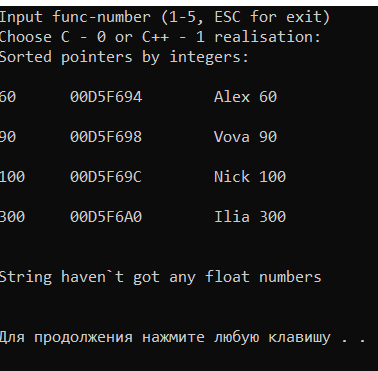


Рисунок 5.5 – Вхідні данні.

  
Рисунок 5.6 – Тестування програми.

**5.2 Текст програмною мовою С.**

**void** taskFiveC()

{

**FILE**\* filePointer = NULL;

**char** name[] = "sentences.txt";

**char** buffer[256];

**char** \*strArray[ROWS];

**int** number = 0;

**int** iMax = 0; **float** fMax = 0;

**int** k = 0, j = 0, l = 0;

**char**\* intArr[MIN];

**char**\* floatArr[MIN];

**char**\* strPointer;

**int** intArray[MIN]; **float** floatArray[MIN];

**int** count;

filePointer = **fopen**(name, "r");

**for** (**int** i = 0; i < MIN; i++)

{

intArray[i] = NULL;

floatArray[i] = NULL;

}

**if** (!filePointer)

{

**perror**("File cannot be open. Try again!");

}

**else**

{

**for** (count = 0; !**feof**(filePointer); count++)

{

**fgets**(buffer, 256, filePointer);

strArray[count] = (**char**\*)**malloc**(**sizeof**(**char**) \* BUFFLEN);

**memcpy**(strArray[count], buffer, 256);

}

}

**fclose**(filePointer);

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

{

strPointer = **strdup**(strArray[i]);

**char**\* token = **strtok**(\*(strArray+i), " ");

**while** (token != NULL)

{

**if** (**isdigit**(\*token))

{

**for** (**int** i = 0; i < **strlen**(token); i++)

{

**if** (token[i] == '.')

number = 1;

}

**if** (number)

{

**if** (fMax < **atof**(token))

fMax = **atof**(token);

number = 0;

}

**else**

{

**if** (iMax < **atoi**(token))

iMax = **atoi**(token);

}

}

token = **strtok**(NULL, " ");

}

**if** (iMax)

{

intArray[k] = iMax;

intArr[k] = strPointer;

k++;

iMax = 0;

}

**if** (fMax)

{

floatArray[j] = fMax;

floatArr[j] = strPointer;

j++;

fMax = 0;

}

}

**if** (k)

{

**printf**("Sorted pointers by integers: \n\n");

**if** (k > 1)

{

**for** (**int** i = 0; intArray[i + 1]; i++)

{

**if** (intArray[i + 1] < intArray[i])

{

**int** iTemp = intArray[i];

intArray[i] = intArray[i + 1];

intArray[i + 1] = iTemp;

**char**\* cTemp = intArr[i];

intArr[i] = intArr[i + 1];

intArr[i + 1] = cTemp;

}

}

**for** (**int** i = 0; intArray[i]; i++)

{

**printf**("%d\t", \*(intArray + i));

**printf**("%p\t",intArr + i);

**printf**("%s\n", \*(intArr + i));

}

}

**else**

{

**printf**("%d\t", \*intArray);

**printf**("%p\t", intArr);

**printf**("%s\n", \*(intArr));

}

}

**else**

{

**printf**("String haven`t got any int numbers\n\n");

}

**if** (j)

{

**printf**("\nSorted pointers by floats: \n\n");

**if** (j > 1)

{

**for** (**int** i = 0; floatArray[i + 1]; i++)

{

**if** (floatArray[i + 1] > floatArray[i])

{

**float** fTemp = floatArray[i];

floatArray[i] = floatArray[i + 1];

floatArray[i + 1] = fTemp;

**char**\* cTemp = floatArr[i];

floatArr[i] = floatArr[i + 1];

floatArr[i + 1] = cTemp;

}

}

**for** (**int** i = 0; floatArray[i]; i++)

{

**printf**("%g\t", \*(floatArray + i));

**printf**("%p\t", floatArr + i);

**printf**("%s\n", \*(floatArr + i));

}

}

**else**

{

**printf**("%g\t", \*floatArray);

**printf**("%p\t", floatArr);

**printf**("%s\n", \*(floatArr));

}

}

**else**

{

**printf**("String haven`t got any float numbers\n\n");

}

**for** (**int** i = 0; i < ROWS; i++) { **free**(\*(strArray + i)); }

}

**Реалізація на мові програмування С++**

**5.3 Тестування програми**

1. Для перевірки вставимо у файл:

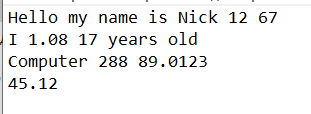


Рисунок 5.7 – Тестування програми.

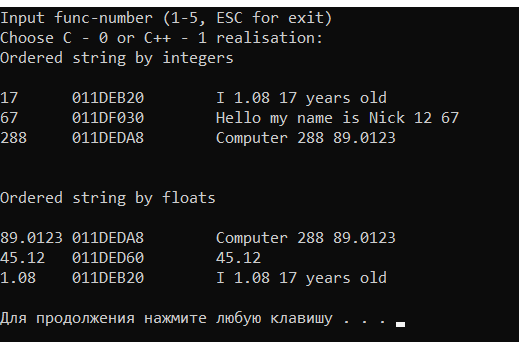


Рисунок 5.8 – Тестування програми.

1. Для перевірки вставимо у файл:

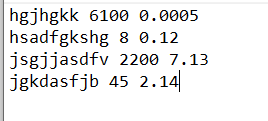


Рисунок 5.9 – Вхідні дані

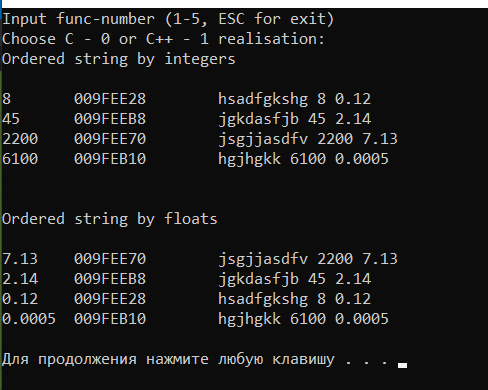


Рисунок 5.10 – Тестування програми.

1. Для перевірки вставимо у файл:

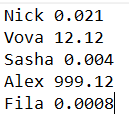


Рисунок 5.11 – Вхідні дані

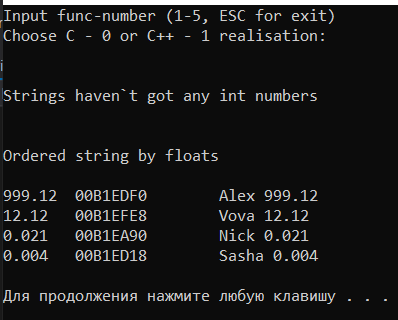


Рисунок 5.12 – Тестування програми.

**5.4 Текст програмною мовою С++.**

**void** taskFiveCPP()

{

**string** path = "sentences.txt";

**ifstream** fin;

fin.**open**(path);

**string** buffer;

**string**\* ptrIntArray[ROWS];

**string**\* ptrFloatArray[ROWS];

**string** strArray[ROWS];

**float** floatMax = 0; **int** intMax = 0;

**float** floatArray[ROWS]; **int** intArray[ROWS];

**int** k = 0, j = 0;

**for** (**int** i = 0; i < ROWS; i++)

{

floatArray[i] = 0;

intArray[i] = 0;

}

**if** (!fin.**is**\_**open**())

{

cout << "Error! File can`t be open. Try again" << **endl**;

}

else

{

**int** i = 0;

**while** (!fin.**eof**())

{

**getline**(fin, buffer);

strArray[i] = buffer;

i++;

}

}

fin.**close**();

**for** (**int** i = 0; i < ROWS; i++)

{

intMax = 0;

floatMax = 0;

**stringstream** ss;

ss << strArray[i];

**string** temp;

**float** foundfloat;

**string**\* strPointer = **new** **string**(strArray[i]);

**while** (!ss.**eof**())

{

ss >> temp;

**if** (stringstream(temp) >> foundfloat)

{

if (foundfloat == trunc(foundfloat))

{

if (foundfloat > intMax)

intMax = foundfloat;

}

**else**

{

if (foundfloat > floatMax)

floatMax = foundfloat;

}

}

}

**if** (intMax)

{

intArray[k] = intMax;

ptrIntArray[k] = strPointer;

k++;

}

**if** (floatMax)

{

floatArray[j] = floatMax;

ptrFloatArray[j] = strPointer;

j++;

}

}

**if** (k)

{

cout << "Ordered string by integers\n\n";

**if** (k > 1)

{

**for** (**int** j = 0; j < 3; j++)

{

**for** (**int** i = 0; intArray[i + 1]; i++)

{

**if** (intArray[i] > intArray[i + 1])

{

**int** itemp = intArray[i];

intArray[i] = intArray[i + 1];

intArray[i + 1] = itemp;

**string**\* stemp = ptrIntArray[i];

ptrIntArray[i] = ptrIntArray[i + 1];

ptrIntArray[i + 1] = stemp;

}

}

}

for (int i = 0; intArray[i]; i++)

{

cout << intArray[i] << '\t' << ptrIntArray[i] << '\t' << \*(ptrIntArray[i]) << **endl**;

}

}

**else**

{

cout << intArray[0] << '\t' << ptrIntArray[0] << '\t' << \*(ptrIntArray[0]) << **endl**;

}

}

**else**

{

cout << "\n\nStrings haven`t got any int numbers" << **endl**;

}

**if** (j)

{

cout << "\n\nOrdered string by floats \n\n";

**if** (j > 1)

{

**for** (**int** j = 0; j < 3; j++)

{

**for** (**int** i = 0; floatArray[i + 1]; i++)

{

**if** (floatArray[i + 1] > floatArray[i])

{

float itemp = floatArray[i];

floatArray[i] = floatArray[i + 1];

floatArray[i + 1] = itemp;

string\* stemp = ptrFloatArray[i];

ptrFloatArray[i] = ptrFloatArray[i + 1];

ptrFloatArray[i + 1] = stemp;

}

}

}

**for** (**int** i = 0; floatArray[i]; i++)

{

cout << floatArray[i] << '\t' << ptrFloatArray[i] << '\t' << \*(ptrFloatArray[i]) << **endl**;

}

}

else

{

cout << floatArray[0] << '\t' << ptrFloatArray[0] << '\t' << \*(ptrFloatArray[0]) << **endl**;

}

}

else

{

cout << "\n\nStrings haven`t got any float numbers" << **endl**;

}

}