Week 09

=====================================================

Problem 901:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// create fucntion that will check if the substring "AITU" can be found in the given line.

void output(char\* stroka, int size, char\* word)

{

int count = 0;

int x = 0;

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

{

if (stroka[i] == word[x])

for (int j = i; j < i + 4; j++, x++)

if (stroka[j] == word[x])

count++;

if (count != 4)

x = 0;

}

(count == 4) ? cout << "YES" : cout << "NO";

}

// create fucntion which convert a -> A

char\* up(char\* massiv)

{

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

(97 <= massiv[i] && massiv[i] <= 122) ? massiv[i] = massiv[i] - 32 : massiv[i];

return massiv;

}

int main()

{

char massiv[100];

char word[] = "AITU";

cin.getline(massiv, 100);

int size = strlen(massiv);

char\* answer = up(massiv);

output(answer, size, word);

}

=====================================================

Problem 902:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// функция которая меняет все элементы в массиве на точку

void replace(char\* text)

{

for (int i = 0; i < stein(text); i++)

if (text[i] == '!')

text[i] = '.';

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

cout << text[i];

}

// точка входа в программу

int main()

{

char text[1000] = "";

cin.getline(text, 1000);

replace(text);

}

=====================================================

Problem 903:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// функция которая разделяет числа по 3 цифры пробелами

void output(int\* stroka, int size)

{

//определяю первые цифры числа которое нельзя сгруппировать по 3

int devide = size % 3;

int probel = 0;

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

cout << stroka[i];

if (devide != 0)

cout << " ";

//вывожу остальные числа с конца

for (int i = devide; i < size; i++)

{

probel++;

cout << stroka[i];

if (probel == 3 && i != size - 1)

{

probel = 0;

cout << " ";

}

}

}

int main()

{

char massiv[100];

int answer[100];

cin.getline(massiv, 100);

int size = strlen(massiv);

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

answer[i] = massiv[i] - '0';

output(answer, size);

}

=====================================================

Problem 904:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

//функция которая находит и возрвращает последнее слово в строке

void output(char\* stroka, int size)

{

int index\_probel = 0;

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

if (stroka[i] == ' ')

index\_probel = i;

for (int i = index\_probel + 1; i < size; i++)

(97 <= stroka[i] && stroka[i] <= 122) ? stroka[i] = stroka[i] - 32 : stroka[i];

for (int i = index\_probel + 1; i < size; i++)

if (65 <= stroka[i] && stroka[i] <= 90)

cout << stroka[i];

}

int main()

{

char massiv[100];

cin.getline(massiv, 100);

int size = strlen(massiv);

output(massiv, size);

}

=====================================================

Problem 905:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

//функция которая преобразует массив символов в массив слов

string\* devide(char\* words, string\* massiv, int count)

{

int index = 0;

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

{

for (int j = index; j < strlen(words); j++)

{

if (words[j] != ' ')

{

massiv[i] += words[j];

index++;

}

else

{

j = strlen(words) - 1;

index++;

}

}

}

return massiv;

}

//функция которая проверяет количество слова в массиве слов и корректирует массив

string\* function(string\* massiv, string word, int count)

{

int counter = 0;

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

if (massiv[i] == word)

{

counter++;

if (counter > 1)

massiv[i] = " ";

}

return massiv;

}

int main()

{

string massiv[35];

char stroka[1000];

cin.getline(stroka, 1000);

int count = 0;

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

if (stroka[i] == ' ')

count++;

count++;

devide(stroka, massiv, count);

//делим массив символов что бы проверить каждое слово

char\* ptr = strtok(stroka, " ");

int result, result\_special;

while (ptr != NULL)

{

function(massiv, ptr, count);

ptr = strtok(NULL, " ");

}

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

{

if (massiv[i] == " ")

continue;

else

{

cout << massiv[i];

if (i != count - 1)

cout << " ";

}

}

}

=====================================================

Problem 906:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

// функция которая проверяет является ли символ цифрой

bool check(char symbol)

{

if ('0' <= symbol && symbol <= '9')

return true;

else

return false;

}

// функция которая разделяет числа по 3 цифры пробелами

void output(int\* stroka, int size)

{

int devide = size % 3;

int probel = 0;

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

cout << stroka[i];

if (devide != 0)

cout << " ";

for (int i = devide; i < size; i++)

{

probel++;

cout << stroka[i];

if (probel == 3 && i != size - 1)

{

probel = 0;

cout << " ";

}

}

}

int main()

{

char massiv[1000];

cin.getline(massiv, 1000);

int size = strlen(massiv);

int count = 0;

int size\_of\_long = 0;

int index = 0;

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

{

if (check(massiv[i]))

massiv[i] = massiv[i];

else

massiv[i] = ' ';

}

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

{

if (massiv[i] != ' ')

{

count++;

size\_of\_long = max(size\_of\_long, count);

index = i;

}

else

count = 0;

}

int k = 0;

char\* total = new char[size\_of\_long];

int\* answer = new int[size\_of\_long];

for (int i = index - size\_of\_long + 1; i < index + 1; i++)

{

total[k] = massiv[i];

k++;

}

int n = size\_of\_long;

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

answer[i] = total[i] - '0';

output(answer, n);

}

=====================================================

Problem 907:

#include <iostream>

#include <string>

using namespace std;

//если строки равны нахожу строку с наибольшим значением

int convert(string number1, string number2, string number3)

{

int index, max\_n = 0;

for (int i = 0; i < number1.length(); i++)

{

if (number1[i] == number2[i] && number2[i] == number3[i])

continue;

else

{

if (number1[i] > number2[i] && number2[i] > number3[i])

return 0;

else if (number1[i] < number2[i] && number2[i] > number3[i])

return 1;

if (number1[i] < number2[i] && number2[i] < number3[i])

return 2;

break;

}

}

}

// находим самую длинную строку

int function(string\* massiv)

{

int size1 = massiv[0].length();

int size2 = massiv[1].length();

int size3 = massiv[2].length();

int index, max\_n = 0;

// если длины строк равны находим наибольшую строку

if (size1 == size2 && size2 == size3)

index = convert(massiv[0], massiv[1], massiv[2]);

else

for (int i = 0; i < 3; i++)

if (massiv[i].length() > max\_n)

{

max\_n = massiv[i].length();

index = i;

}

return index;

}

int main()

{

string numbers[3];

for (int i = 0; i < 3; i++)

cin >> numbers[i];

cout << numbers[function(numbers)];

}

=====================================================

=====================================================

Problem 908:

=====================================================

#include <iostream>

#include <string>

#include<bits/stdc++.h>

using namespace std;

string Summa(string\* massiv)

{

if (massiv[0].length() > massiv[1].length())

swap(massiv[0], massiv[1]);

string str = "";

int n1 = massiv[0].length(), n2 = massiv[1].length();

reverse(massiv[0].begin(), massiv[0].end());

reverse(massiv[1].begin(), massiv[1].end());

int carry = 0;

for (int i = 0; i < n1; i++)

{

int sum = ((massiv[0][i] - '0') + (massiv[1][i] - '0') + carry);

str.push\_back(sum % 10 + '0');

carry = sum / 10;

}

for (int i = n1; i < n2; i++)

{

int sum = ((massiv[1][i] - '0') + carry);

str.push\_back(sum % 10 + '0');

carry = sum / 10;

}

if (carry)

str.push\_back(carry + '0');

reverse(str.begin(), str.end());

return str;

}

int main()

{

string massiv[2];

for (int i = 0; i < 2; i++)

cin >> massiv[i];

cout << Summa(massiv);

return 0;

}

=====================================================

Problem 909:

=====================================================

#include <iostream>

#include <string>

using namespace std;

//функция которая проверяет делится ли число на 11 или нет

// если разность суммы нечетных и суммы четных делится на 11 тогда вернется правда

int check(string str, int n)

{

int ODD = 0, EVEN = 0;

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

(i % 2 == 0) ? ODD += (str[i] - '0') : EVEN += (str[i] - '0');

return ((ODD - EVEN) % 11 == 0);

}

int main()

{

string str;

cin >> str;

int n = str.length();

check(str, n) ? cout << "YES" : cout << "NO";

return 0;

}

=====================================================

Problem 910:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

int count = 0, i = 0;

char s[10000];

cin.getline(s, 10000);

int n = strlen(s);

//пока не конец строки я сравнивая элементы и нахожу количество нужных

while (s[i] != '\0')

{

if (s[i] == '>' && s[i + 1] == '>' && s[i + 2] == '-' && s[i + 3] == '-' && s[i + 4] == '>')

count++;

if (s[i] == '<' && s[i + 1] == '-' && s[i + 2] == '-' && s[i + 3] == '<' && s[i + 4] == '<')

count++;

i++;

}

cout << count;

return 0;

}

=====================================================

Problem 911:

=====================================================

#include <iostream>

using namespace std;

int main()

{

int column, row;

cin >> column >> row;

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

{

if ((i % 2 == 0 && column % 2 == 0) || (i % 2 == 1 && column % 2 == 1))

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

(j % 2 == 0) ? cout << 'X' : cout << '.';

else if ((i % 2 == 0 && column % 2 == 1) || (i % 2 == 1 && column % 2 == 0))

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

(j % 2 == 0) ? cout << '.' : cout << 'X';

cout << endl;

}

}

=====================================================

Problem 912:

=====================================================

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

char stroka[101]{ "\0" };

char stoka[101]{ "\0" };

cin.getline(stroka, 101);

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

{

if (stroka[i] == '#')

break;

stoka[i] = stroka[i];

}

// используя strtok я делю нахожу только слова которые мне нужны

char\* ptr = strtok(stoka, " ,.-?0!123456789");

long double count = 0, total = 0, answer;

while (ptr != NULL)

{

total += strlen(ptr);

count++;

ptr = strtok(NULL, " ,.-?0!123456789");

}

if (count != 0)

answer = total / count;

else

answer = 0;

//вывод с точностью до 10^-9

cout.setf(ios::fixed);

cout.precision(9);

cout << answer;

}