Отчет по практике второго семестра C++  
по дисциплине

«Информатика и программирование»

Подготовил студент I курса  
СГУ, факультет КНиИТ  
Акимов Артемий Андреевич  
Руководитель  
Портенко Марина Сергеевна

Саратов 2013 г.

**Структуры I.18**

Найти три различные точки из заданного множества точек, образующих  
треугольник наибольшею периметра.

#include <iostream>

#include <cmath>

#include <stdio.h>

using namespace std ;

struct toc

{

double x,y;

};

double pl(toc a,toc b,toc c)

{

double e , w, q,p;

e=sqrt((a.x-b.x)\*(a.x-b.x)+(a.y-b.y)\*(a.y-b.y));

w=sqrt((b.x-c.x)\*(b.x-c.x)+(b.y-c.y)\*(b.y-c.y));

q=sqrt((c.x-a.x)\*(c.x-b.x)+(c.y-a.y)\*(c.y-a.y)) ;

p=(e+w+q)/2;

return sqrt(p\*(p-e)\*(p-w)\*(p-q));

}

int main()

{

freopen("input.txt","r",stdin);

freopen("output.txt","w",stdout);

toc mas[100];

int n ;

cin>>n;

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

{

cin>>mas[i].x>>mas[i].y;

}

double max=-1000;

toc ans[3];

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

{

ans[i].x=0;

ans[i].y=0;

}

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

{

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

{

for(int r = j+1 ; r< n; r++)

{

if(max<pl(mas[i],mas[j],mas[r]))

{

ans[0].x=mas[i].x;

ans[0].y=mas[i].y;

ans[1].x=mas[r].x;

ans[1].y=mas[r].y;

ans[2].x=mas[j].x;

ans[2].y=mas[j].y;

max= pl(mas[i],mas[j],mas[r]) ;

}

}

}

}

cout<<max;

return 0 ;

}

**Структуры II.10**

**А)**

На основе данных входного файла составить список сотрудников учреждения, включив следующие данные: ФИО. год принятия на работу, должность, зарплата, рабочий стаж. Вывести в новый файл список сотрудников учреждения, удалив из него информацию о сотрудниках, принятых на работу в текущем году.

**Б)**

Отсортировать по должности, ФИО

**A)**

#include<fstream>

#include<string>

#include<iostream>

#include<iomanip>

using namespace std;

ifstream infile("input.txt");

ofstream outfile("output.txt");

struct spisok

{

int god,zarplata,stazh;

string fio,dolzhn;

void show();

void print();

};

void spisok::show()

{

cout<<setw(10)<<fio<<setw(20)<<god<<setw(10)<<dolzhn<<setw(10)<<zarplata<<setw(10)<<stazh<<endl;

};

void spisok::print()

{

outfile<<setw(10)<<fio<<setw(10)<<god<<setw(10)<<dolzhn<<setw(10)<<zarplata<<setw(10)<<stazh<<endl;

};

int main ()

{

spisok vsego[5];

int data=2014, n=0;

for (int i = 0; i < 5; i++)

{

infile>>vsego[i].fio;

infile>>vsego[i].god;

infile>>vsego[i].dolzhn;

infile>>vsego[i].zarplata;

infile>>vsego[i].stazh;

vsego[i].show();

n++;

}

int j=0;

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

{

if (vsego[i].god == data)

{

for(j=i;j<n-1;j++) vsego[j]=vsego[j+1];

n--;

}

}

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

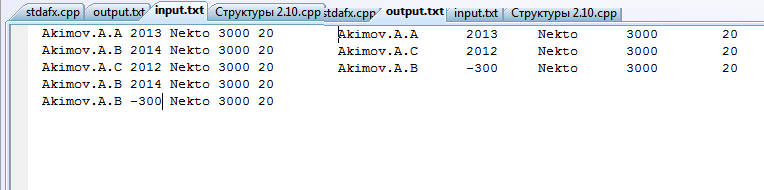
vsego[i].print();

infile.close();

outfile.close();

return 0;

}



**Б)**

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

#include <algorithm>

using namespace std;

ifstream infile("input.txt");

ofstream outfile("output.txt");

struct spisok

{

int god,zarplata,stazh;

string fio,dolzhn;

void show();

void print();

};

void spisok::show()

{

cout<<setw(10)<<fio<<setw(20)<<god<<setw(10)<<dolzhn<<setw(10)<<zarplata<<setw(10)<<stazh<<endl;

};

bool compare (spisok a, spisok b){

return a.dolzhn<b.dolzhn || (a.dolzhn==b.dolzhn && a.fio<b.fio);}

int main ()

{

spisok vsego[5];

int data=2014, n=0; int i=0;

for (i = 0; i < 5; i++)

{

infile>>vsego[i].fio;

infile>>vsego[i].god;

infile>>vsego[i].dolzhn;

infile>>vsego[i].zarplata;

infile>>vsego[i].stazh;

n++;

}

sort (vsego,vsego + i,compare);

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

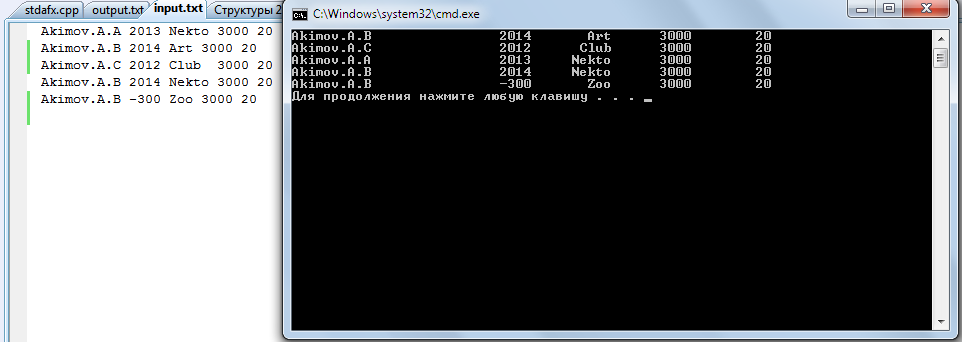
{vsego[j].show();}

infile.close();

outfile.close();

return 0;

}



**Структуры II.16**

На основе данных входного файла составить список вкладчиков банка, включив следующие данные: ФИО. .№ счета, сумма, год открытия счета. Вывести в новый файл информацию о тех вкладчиках, сумма вклада которых превышает заданное значение.

#include<fstream>

#include<string>

#include<iostream>

#include<iomanip>

using namespace std;

ifstream infile("input.txt");

ofstream outfile("output.txt");

struct spisok

{

int nomer, summa, god;

string fio;

void show();

void print();

};

void spisok::show()

{

cout<<setw(20)<<fio<<setw(10)<<nomer<<setw(10)<<summa<<setw(10)<<god<<endl;

};

void spisok::print()

{

outfile<<setw(20)<<fio<<setw(10)<<nomer<<setw(10)<<summa<<setw(10)<<god<<endl;

};

int main ()

{

spisok vsego[20];

int data = 5000, n = 0;

for (int i = 0; i < 10; i++)

{

infile>>vsego[i].fio;

infile>>vsego[i].nomer;

infile>>vsego[i].summa;

infile>>vsego[i].god;

n++;

}

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

{

if (vsego[i].summa > data)

vsego[i].print();

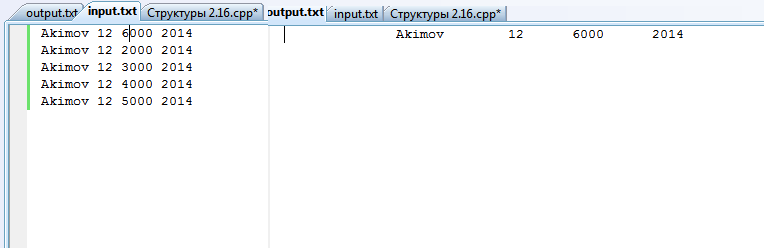
}

infile.close();

outfile.close();

return 0;

}



**Функции 1.1**

Разработка нерекурсивных функций.

Разработать функцию, которая для заданного натурального числа N возвращает значение true - если число простое, false - если число составное. С помощью данной функции:

вывести на экран все простые числа на отрезке (а, b);

#include <iostream>

using namespace std;

bool prostoe\_quest(int N,int b)

{

int sum=0;

for (int i=1; i<=b; i++)

if (N%i==0) sum++;

if (sum==2) return true;

else return false;

}

int main ()

{

setlocale(LC\_ALL, "Russian");

int a, b;

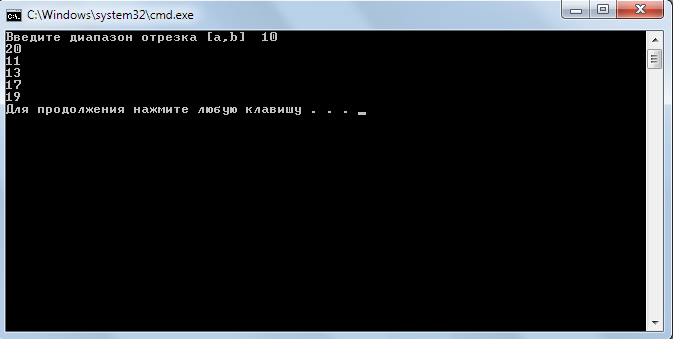
cout<<"Введите диапазон отрезка [a,b]\t"; cin>>a>>b;

for (int N=a; N<=b; N++)

if (prostoe\_quest(N,b)==true) cout<<N<<endl;

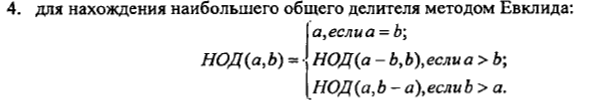
return 0;

}



**Функции 2.4**

Разработать рекурсивную функцию, возвращающую значение.



#include <iostream>

using namespace std;

int rec(int a, int b)

{

if (a==b)

return a;

else if (a>b)

return rec(a-b,b);

else

return rec(a,b-a);

}

int main()

{

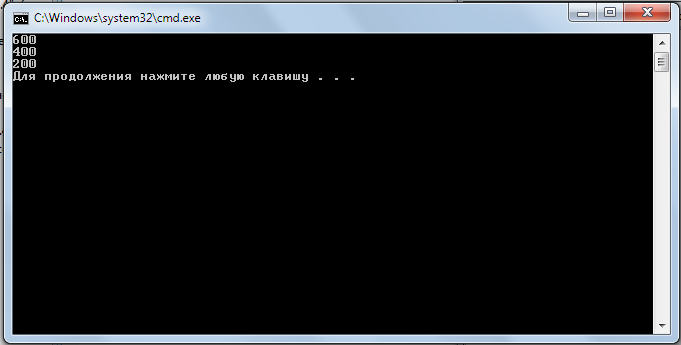
int n,b;

cin>>n;

cin>>b;

cout<<rec(n,b)<<endl;

}

****

**Функции 3.3**

Разработать рекурсивную функцию, которая по заданному натуральному числу N выведет на экран все натуральные числа, не превышающие N, в порядке возрастания, например, для N=8, на экран выводится 1 2 3 4 5 6 7 8.

#include <iostream>

using namespace std;

void recurcy(int a,int n)

{

if (a<=n)

{

cout<<a<<' ';

a++;

recurcy(a,n);

}

}

int main()

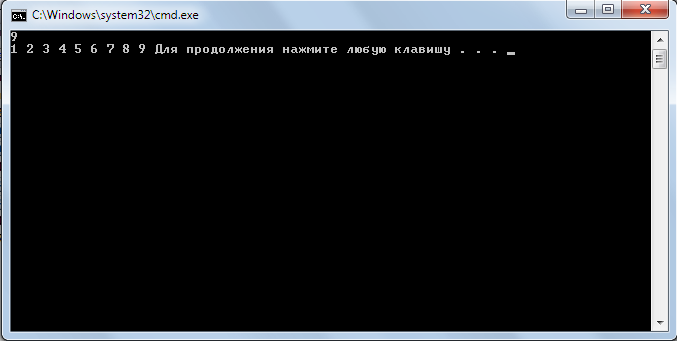
{

int n, a=1;

cin>>n;

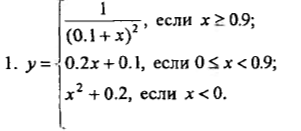
recurcy(a,n);

}

****

**Функции 4.1**

**C:\Users\Artyom\Desktop\Новая папка\Новый точечный рисунок - копия - копия (3).bmp**

****

#include <iostream>

using namespace std;

float F( float x );

void F( float x, float &y );

int main() {

setlocale(LC\_ALL, "Russian");

float x1, x2, y;

cout << "Введите x1 и x2 через пробел" << endl;

cin >> x1 >> x2;

F( x2, y );

cout << "F( " << x1 << " ) = " << F( x1 ) << endl;

cout << "F( " << x2 << " ) = " << y << endl;

return 0;

}

float F( float x ) {

if ( x >=0.9 )

return 1 / ( 0.1 + x\*x );

if ( x < 0 )

return x\*x+0.2;

else

return 1 / (0.2\*x + 0.1 );

}

void F( float x, float &y ) {

if ( x >=0.9 )

y = 1 / ( 0.1 + x\*x );

else

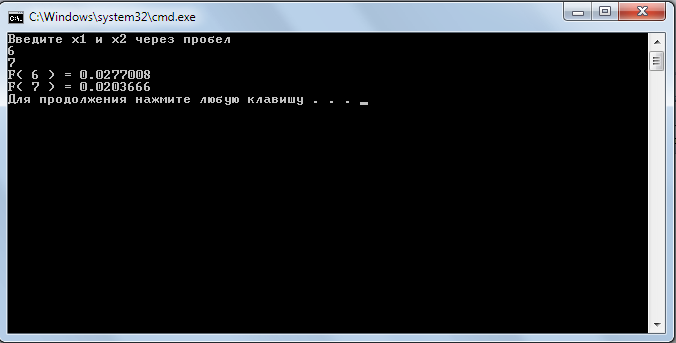
if ( x < 0 )

y = x\*x+0.2;

else

y = 1 / (0.2\*x + 0.1 );

}

****

**Функции 5.3**

Использование функций-шаблонов: для работы с двумерными массивами арифметических типов данных разработать шаблоны ввода и вывода массива, а также шаблон для решения основной задачи:

#include<iostream>

#include <fstream>

using namespace std;

ifstream infile("input.txt");

ofstream outfile("output.txt");

template<typename X>

void printArray(X\*\*mas,int n,int m)

{

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

{

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

outfile<<mas[i][j]<<" ";

outfile<<endl;

}

}

template<typename X>

void Func (X \*\*mas,int n,int m,int a,int b)

{

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

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

if ((mas[i][j]>=a) && (mas[i][j]<=b) )

{

mas[i][j]=0;

}

}

int main()

{ int n,m,a,b;

infile>>n>>m>>a>>b;

int \*\*mas = new int\*[n];

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

mas[i] = new int[m];

{

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

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

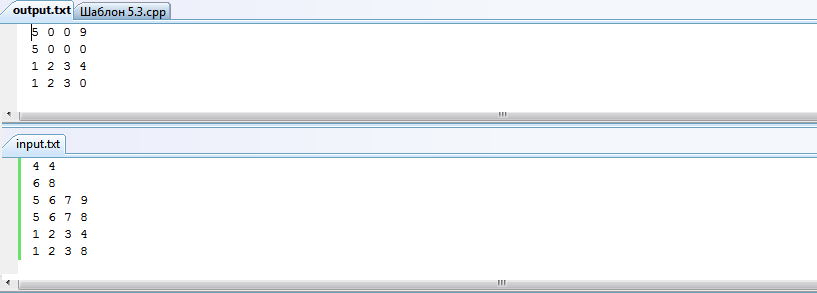
infile>>mas[i][j];

}

Func (mas, n, m, a, b);

printArray(mas,n,m);

}

****

**Файлы 1.11**

Дан текстовый файл. Напечатать все строки, в которых имеется хотя бы один  
пробел.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

ifstream in("input.txt");

ofstream out("output.txt");

string s;

while( !in.eof() )

{

getline(in, s);

if(s.find(' ') != string::npos)

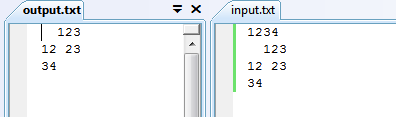
out << s << '\n';

}

in.close();

return 0;

}

****

**Файлы 2.8**

Даны два файла с числами. Получить новый файл, записав в него сначала все  
четные числа из первого файла, потом все нечетные числа из второго.

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main ()

{

ifstream in("input\_chet.txt");

ofstream out ("output.txt");

ifstream sin("input\_nechet.txt");

int i;

while (in.peek()!=EOF)

{

in>>i;

if (i%2==0) out<<i<<" ";

}

out<<"\n";

while (sin.peek()!=EOF)

{

sin>>i;

if (i%2!=0) out<<i<<" ";

}

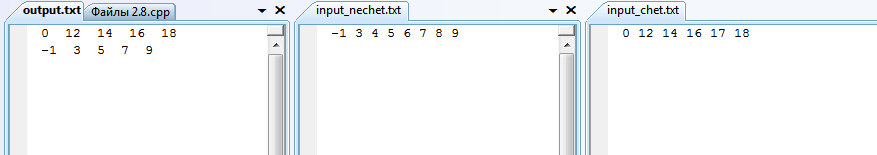
in.close();

sin.close();

out.close();

return 0;

}

****

**Файлы 3.15**

Создать файл, состоящий из n вещественных чисел. Найти максимальное  
значение среди чисел файла, стоящих на нечетных позициях.

#include <iostream>

#include <fstream>

#include <ctime>

using namespace std;

int main()

{

int n;

cin>>n;

ofstream out("outfile.txt", ios::binary);

double mas[30];

for (int i = 1; i <= n; i++)

cin>>mas[i];

out.write((char\*)&mas, sizeof(mas));

out.close();

ifstream in("outfile.txt", ios::binary);

double max = mas[0];

in.read((char \*)& mas, sizeof(mas));

for (int i = 1; i <= n; i++)

{

if ((mas[i] > max) && (i & 1)) (max = mas[i]);

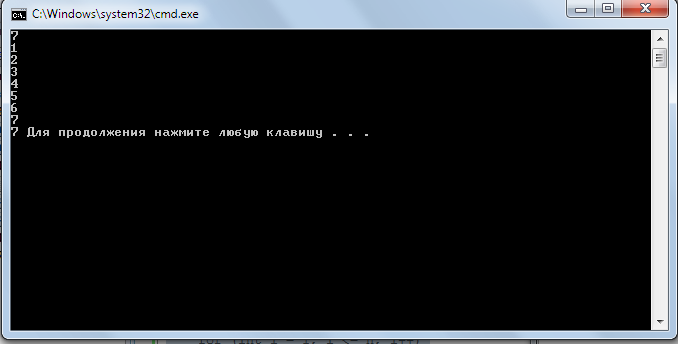
}

cout<<max<<" ";

in.close();

return 0;

}

****

**Сортировка II.4**

Дана матрица размерностью n x n, содержащая целые числа. Отсортировать каждый столбец матрицы по возрастанию элементов методом выбора;

#include <iostream>

#include <stdio.h>

using namespace std ;

void str(int \*\*a , int n,int r ){

int temp, lowindex, lowkey;

int i,j;

for( i=1; i<n; i++)

{

lowindex=i;

lowkey=a[i][r];

for( j=1; j<n; j++)

if (a[j][r]<lowkey)

{

lowkey=a[j][r];

lowindex=j;

}

temp=a[i][r];

a[i][r]=a[lowindex][r];

a[lowindex][r]=temp;

}

}

int main(){

freopen("in.txt","rt",stdin);

freopen("out.txt","wt",stdout);

int n ;

cin>>n;

int \*\*mas = new int\*[n];

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

mas[i]=new int [n];

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

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

cin>> mas[i][j];

}

}

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

str(mas , n , i);

}

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

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

cout<< mas[i][j]<<' ';

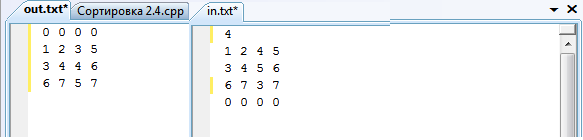
}

cout<<endl;

}

return 0 ;

}



**Контест. Строки. Задача 2**

Во входном файле дана строка [S](http://school.sgu.ru/filter/tex/displaytex.php?S). Строка [S](http://school.sgu.ru/filter/tex/displaytex.php?S) состоит из букв латинского алфавита, непустая и её длина не превышает [100](http://school.sgu.ru/filter/tex/displaytex.php?100) символов.

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int main()

{

ifstream in("input.txt");

ofstream out("output.txt");

int i;

int flag=0;

string str;

in>>str;

for(i=0; i<str.length(); i++)

{

if(str[i]==str[i+1])

{

flag=1;

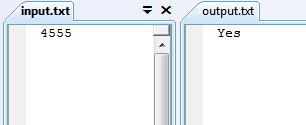
}

}

flag ? out<<"Yes " : out<<"No ";

return 0;

}

****

**Контест. Строки. Задача 8**

Дана строка [S](http://school.sgu.ru/filter/tex/displaytex.php?S), состоящая из слов, разделённых пробелами.   
Выведите все слова из [S](http://school.sgu.ru/filter/tex/displaytex.php?S), которые начинаются и оканчиваются одной буквой.

#include "string"

#include "fstream"

using namespace std;

int main()

{ ifstream cin("input.txt");

ofstream cout("output.txt");

string str;

while (cin>>str)

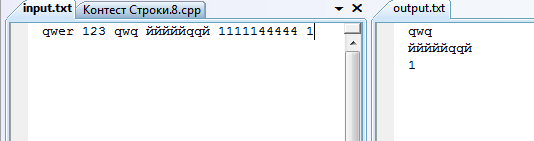
{

if (str[0]==str[str.length()-1]) cout<<str<<endl;

}

return 0;

}

****

**Вектор II.3**

Удалить из массива все числа, значения которых попадают в данный интервал.

#include"iostream"

#include"algorithm"

#include"vector"

#include<conio.h>

using namespace std;

namespace interval

{

int a,b;

}

bool isNeg(int x)

{

return (x >= interval::a && x <= interval::b);

}

int main()

{

vector <int> iVec;

int x,n;

cout<<"n="; cin>>n;

cout<<"a= "; cin>>interval::a;

cout<<"b= "; cin>>interval::b;

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

{

cin>>x;

iVec.push\_back(x);

}

vector<int>::iterator iEnd=remove\_if(iVec.begin(),iVec.end(),isNeg);

iVec.erase(iEnd,iVec.end());

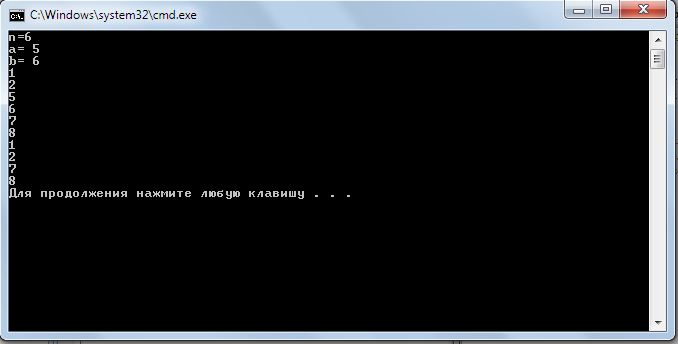
for(vector<int>::iterator iter=iVec.begin();iter<iVec.end();iter++)

cout<<\*iter<<endl;

getch();

return 0;

}

****

**Объект 1.**

Создать класс Triangle, содержащий следующие члены класса:

1. Поля:

* Int a,b,c

1. Функции, позволяющие:

* Вывести на экран информацию о треугольнике;
* Рассчитать периметр треугольник;
* Рассчитать площадь треугольника;
* Установить длины сторон треугольника;
* Установить, существует ли треугольник с данными длинами сторон;
* Перегрузку операций ++(--); одновременно увеличить (уменьшить) значение полей a, b, c на 1
* Перегрузку операции \*: умножить поля a, b, c на заданный скаляр.

#include <iostream>

#include <math.h>

using namespace std;

class Triangle

{

private:

double a;

double b;

double c;

public:

int GetA() { return a;}

int GetB() { return b;}

int GetC() { return c;}

Triangle (): a(1), b(1), c(1)

{}

Triangle (double a, double b, double c):a(a),b(b), c(c)

{}

void Info () {

cout<<"длина стороны a = "<<a<<endl;

cout<<"длина стороны b = "<<b<<endl;

cout<<"длина стороны c = "<<c<<endl;

}

double Perimetr () {

return a+b+c;

}

double Area () {

double p = (a+b+c)/2;

double k = (p\*(p-a)\*(p-b)\*(p-c));

return

pow(k,0.5) ;

}

void SetAB (double a, double b, double c) {

this->a=a;

this->b=b;

this->c=c;

}

bool Existence () {

if ((a+b>c) && (a+c>b) && (c+b>a)) return true;

return false;

}

Triangle operator++() {

return Triangle(++a, ++b, ++c);

}

Triangle operator--() {

return Triangle(--a, --b, --c);

}

Triangle operator\*(int f) {

return Triangle(a\*f, b\*f, c\*f);

}

};

int main()

{

setlocale (LC\_ALL, "Rus");

Triangle TriangleA;

TriangleA.Info();

if (TriangleA.Existence())

{

cout<<"Площадь = "<<TriangleA.Area()<<endl;

}

cout<<"Периметр = "<<TriangleA.Perimetr()<<endl<<endl;

TriangleA.SetAB(3,4,5);

TriangleA.Info();

if (TriangleA.Existence())

{

cout<<"Площадь = "<<TriangleA.Area()<<endl;

}

cout<<"Периметр = "<<TriangleA.Perimetr()<<endl<<endl;

if (TriangleA.Existence()) cout<<"треугольник со сторонами ("<<TriangleA.GetA()<<","<<TriangleA.GetB()<<","<<TriangleA.GetC()<<") существует"<<endl;

Triangle TriangleB(3,5,6);

TriangleB.Info();

if (TriangleB.Existence())

{

cout<<"Площадь = "<<TriangleB.Area()<<endl;

}

cout<<"Периметр = "<<TriangleB.Perimetr()<<endl<<endl;

++TriangleB;

TriangleB.Info();

if (TriangleB.Existence())

{

cout<<"Площадь = "<<TriangleB.Area()<<endl;

}

cout<<"Периметр = "<<TriangleB.Perimetr()<<endl<<endl;

--TriangleB;

TriangleB.Info();

if (TriangleB.Existence())

{

cout<<"Площадь = "<<TriangleB.Area()<<endl;

}

cout<<"Периметр = "<<TriangleB.Perimetr()<<endl<<endl;

Triangle TriangleC=TriangleB\*3;

TriangleC.Info();

if (TriangleC.Existence())

{

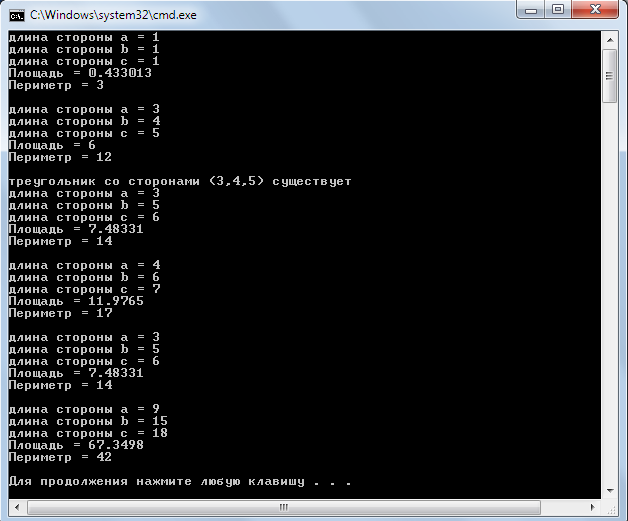
cout<<"Площадь = "<<TriangleC.Area()<<endl;

}

cout<<"Периметр = "<<TriangleC.Perimetr()<<endl<<endl;

return 0;

}



**Контест. Стеки. Задача 3**

Дан текст. Переписать его, выписывая символы каждой строки в обратном порядке. Решение должно основываться на использовании структуры данных стек.

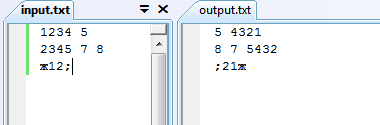
#include <iostream>

#include <stack>

#include <stdio.h>

#include <string>

ГОТОрОЫ) .11: .4 ука-НТСЛЬ Ml ОбъОП LVb ■ того гит

содержи ссылку об\*еп ■ ••■■р.. псыоЛ куче или равен mil На****

**Контест. Очередь. Задача 3**

Дан файл состоящий из слов, разделенных одним или несколькими пробелами, табуляциями и переводами строки. Словом является последовательность символов состоящая из больших и маленьких латинских букв, состоящая из не более чем 20 символов.   
Необходимо напечатать сперва все слова начинающиеся на гласную букву, а затем все слова начинающиеся на согласную букву, сохраняя исходный порядок в каждой группе слов. Гласными буквами в английском языке являются буквы [a, e, i, o, u, y](http://school.sgu.ru/filter/tex/displaytex.php?a,+e,+i,+o,+u,+y).   
(Рекомендуется решать задачу, используя очередь)

#include <iostream>

#include <queue>

#include <string>

#include <fstream>

using namespace std;

ifstream in("input.txt");

ofstream out("output.txt");

using namespace std;

int main()

{

queue <string> que;

string s;

while(in >> s)

{ char c = s[0];

if (s[0]=='a' || s[0]=='e' || s[0]=='o' || s[0]=='u' || s[0]=='y' || s[0]=='i' || s[0]=='A' || s[0]=='O' || s[0]=='U' || s[0]=='Y' || s[0]=='I'|| s[0]=='E' )

{

out<<s<<" ";

}

else

{

que.push(s);

}

}

while (!que.empty()) {

out << que.front() << ' ';

que.pop();

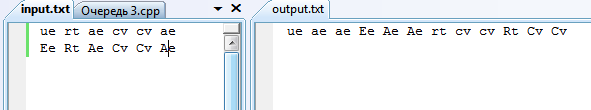
}

in.close();

out.close();

return 0;

}

****