Задание №1

Условие:

Задана матрица А с 3 столбцами и 25 строками. Каждая строка матрицы содержит длины сторон одного из 25 треугольников. Определить номера тех строк, которые содержат длины сторон, не допускающие построение треугольника.

Алгоритм процедуры generate:

Алгоритм процедуры initarray:

Алгоритм процедуры enter:

Алгоритм процедуры file\_in:

Алгоритм процедуры monitor\_out:

Алгоритм процедуры outarray:

Алгоритм процедуры counter:

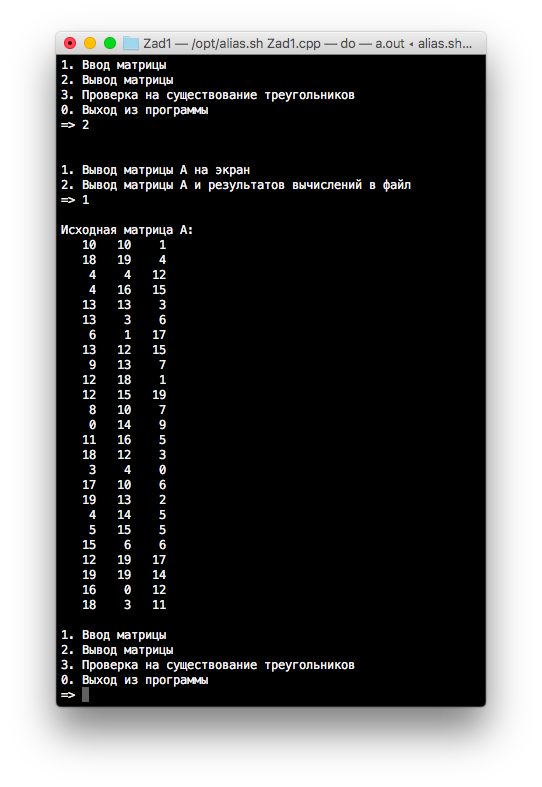
Алгоритм процедуры file\_out:

Алгоритм основной программы:

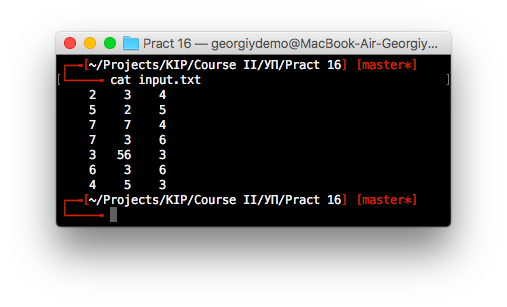
Исходный код программы:

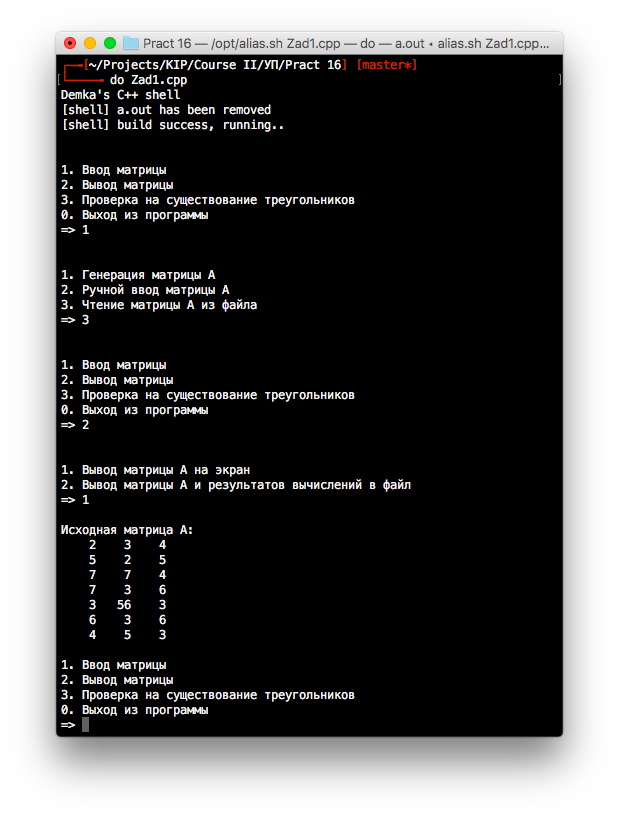
#include **<iostream>**#include **<iomanip>**#include **<fstream>  
using namespace** std;  
  
**int const** n = 25;  
**int const** m = 3;  
  
**int** i, j, k1, k2, menu, \*\*a;  
FILE \*outfile;  
  
*//Генерация массивов***void** generate(){  
  
 srand((**unsigned**)time(0));  
 **for** (i=0;i<n;i++)  
 **for** (j=0;j<m;j++)  
 a[i][j]=rand()%20;  
  
}  
  
*//Ручный ввод массивов***void** enter(){  
  
 cout<<**"<Заполняем матрицу А>\n"**;  
 **for** (i=0;i<n;i++)  
 **for** (j=0;j<m;j++){  
 cout<<**"Введите элемент ["**<<i<<**"]["**<<j<<**"]: "**;  
 cin>>a[i][j];  
 }  
  
}  
  
*//Ввод массивов из файла***void** file\_in(){  
  
 fstream fi;  
 fi.open(**"input.txt"**);  
 **for** (i=0;i<n;i++)  
 **for** (j=0;j<m;j++)  
 fi>>a[i][j];  
 fi.close();  
  
}  
  
*//Вывод массивов на экран***void** monitor\_out(){  
  
 cout<<**"\nИсходная матрица А:"**;  
 **for** (i=0;i<n;i++){  
 cout<<**"\n"**;  
 **for** (j=0;j<m;j++)  
 cout<<setw(5)<<a[i][j];  
 }  
}  
  
*//Меню выбора ввода***void** initarray(){  
  
 a = **new int** \*[n];  
 **for**(i=0;i<n;i++)  
 a[i]=**new int**[m];  
  
 cout<<**"\n\n1. Генерация матрицы А\n2. Ручной ввод матрицы А\n3. Чтение матрицы А из файла\n=> "**;  
 cin>>k1;  
 **switch** (k1)  
 {  
 **case** 1:  
 generate();  
 **break**;  
  
 **case** 2:  
 enter();  
 **break**;  
  
 **case** 3:  
 file\_in();  
 **break**;  
 }  
  
}  
  
*//Процедура формирования массива x по образцу***void** counter(){  
   
 **for** (i=0;i<n;i++){  
 **if** ((a[i][0]<(a[i][1]+a[i][2])) && (a[i][1]<(a[i][0]+a[i][2])) && (a[i][2]<(a[i][1]+a[i][0])))  
 printf(**"\n✅ Строка №%d: треугольник существует [%d, %d, %d]"**,i+1,a[i][0],a[i][1],a[i][2]);  
 **else** printf(**"\n❌ Строка №%d: треугольник не существует [%d, %d, %d]"**,i+1,a[i][0],a[i][1],a[i][2]);  
 }  
  
}  
  
*//Процедура вывода в файл***void** file\_out(){  
  
 outfile=fopen(**"out.txt"**,**"w+"**);  
 fprintf(outfile,**"Матрица А:\n"**);  
 **for** (i=0;i<n;i++){  
 **for** (j=0;j<m;j++)  
 fprintf(outfile,**"%4d"**,a[i][j]);  
 fprintf(outfile,**"\n"**);  
 }  
   
 **for** (i=0;i<n;i++){  
 **if** ((a[i][0]<(a[i][1]+a[i][2])) && (a[i][1]<(a[i][0]+a[i][2])) && (a[i][2]<(a[i][1]+a[i][0])))  
 fprintf(outfile,**"\n✅ Строка №%d: треугольник существует [%d, %d, %d]"**,i+1,a[i][0],a[i][1],a[i][2]);  
 **else** fprintf(outfile,**"\n❌ Строка №%d: треугольник не существует [%d, %d, %d]"**,i+1,a[i][0],a[i][1],a[i][2]);  
   
 }  
   
 fprintf(outfile,**"\n"**);  
 fclose(outfile);  
  
}  
  
*//Меню выбора вывода***void** outarray(){  
  
 cout<<**"\n\n1. Вывод матрицы A на экран\n2. Вывод матрицы А и результатов вычислений в файл\n=> "**;  
 cin>>k2;  
 **switch** (k2)  
 {  
 **case** 1:  
 monitor\_out();  
 **break**;  
  
 **case** 2:  
 file\_out();  
 **break**;  
 }  
  
}  
  
**int** main(){  
  
 **do** {  
 cout<<**"\n\n1. Ввод матрицы\n2. Вывод матрицы\n3. Проверка на существование треугольников\n0. Выход из программы\n=> "**;  
 cin>>menu;  
 **switch** (menu)  
 {  
 **case** 1:  
 initarray();  
 **break**;  
  
 **case** 2:  
 outarray();  
 **break**;  
  
 **case** 3:  
 counter();  
 **break**;  
  
 }  
  
 }  
 **while** (menu!=0);  
  
cout<<**"\n"**;  
**return** 0;  
}

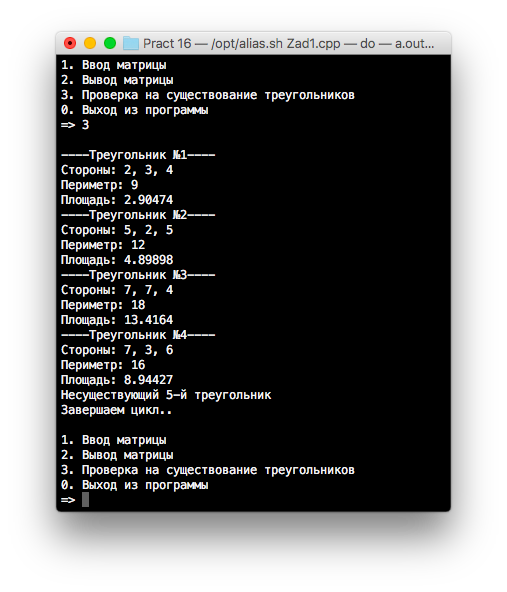
Скриншоты программы:

а) Ввод исходной матрицы через генератор случайных чисел и ее вывод на экран

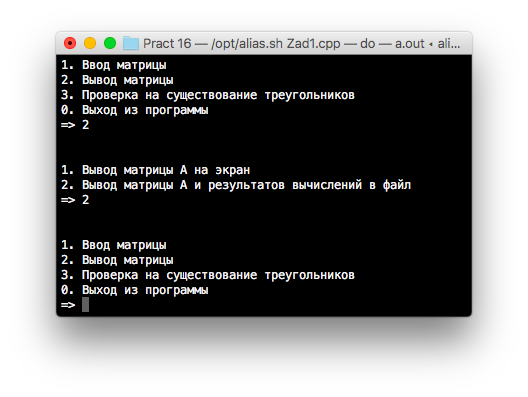
в) Исходный файл с матрицей



б) Чтение матрицы из файла и ее вывод на экран

д) Проверка на существование треугольников из элементов строк матрицы, согласно условию

е) Вывод результатов всех операций в файл



ё) Исходный файл на выходе

