**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Инженерная академия

*Департамент механики и мехатроники*

Курсовая работа

**по дисциплине "Программирование и основы алгоритмизации"**

*на тему*

«Объектно-ориентированное программирование»

*(Вариант № 24-45)*

Выполнил: Лебедев Д.М.

Группа: ИУБ-102

Страна: Россия

Проверил:

Оценка:

Дата:

**Москва 2018 г.**

# Оглавление

1. Оглавление 2
2. Анализ задания 3
3. Пример работы приложения 5
4. Листинг кода 6

**Анализ задания**

24. Найти в заданной последовательности чисел подмножество из k чисел, сумма которых даёт наибольшее число, из множества {ai = ai-1 + i, a0 = 0; i = 0,1, …};

45. Сортировка выбором по возрастанию.

В данном задании требуется найти такое подмножество из k (число k вводится пользователем) элементов, сумма членов которого даёт одно из чисел из известного множества {ai = ai-1 + i, a0 = 0; i = 0,1, …}. Причём при наличии нескольких таких подмножеств выводиться пользователю должно то, элементы которого в сумме дают большее число.

Решение поставленной задачи начинается с создания таблицы StringGrid, в ячейки которой впоследствии будут записываться члены последовательности (элементы рандомного массива, которых, кстати, не может быть больше 30 из-за слишком большого времени расчётов), элементы известного множества {ai = ai-1 + i, a0 = 0; i = 0,1, …}, искомое подмножество и сумма его членов. Также с помощью Edit задаются диапазон возможных значений членов последовательности и их количество. В свою очередь, вывод всех нужных последовательностей происходит с помощью кнопок (Button), которые:

1) Если являются первостепенными, то ради удобства находятся над соответствующими столбцами ячеек StringGrid’a. Button “Crear” - над нулевым столбцом; “Formula” – над первым столбцом; “Secuencia” – над вторым столбцом и “Suma” – над третьим столбцом.

2) Если являются вспомогательными, то находятся под StringGrid’ом. Кнопки “Sort”, “Mix” и “Descargar”.

После нажатия на “Crear” (исп. “Создать”) в зависимости от диапазона значений и количества элементов выводится случайный массив. В то же самое время, он записывается в текстовый файл.

После нажатия на “Formula” в зависимости от количества элементов случайным образом созданного множества выводятся в StringGrid элементы известного множества {ai = ai-1 + i, a0 = 0; i = 0,1, …} – сколько элементов в рандомном множестве, столько же будет показано в StringGrid’е элементов известного множества.

После нажатия на “Secuencia” (исп. “Последовательность”) выводится как в StringGrid, так и в текстовый файл, искомая последовательность: та, элементы которой в сумме дают один из элементов множества {ai = ai-1 + i, a0 = 0; i = 0,1, …}. Как было сказано выше, если таковых последовательностей найдётся несколько, то выведется та, сумма элементов которой даёт большее число из множества. А если существуют несколько множеств с одинаковыми суммами, то выведется то, индексы элементов которого находятся ближе к нулю. Также под “Suma” (исп. “Сумма”) станет видной конечная сумма элементов искомой последовательности.

После нажатия на “Sort” произойдет сортировка случайного массива, выведенного после нажатия на “Crear”, методом выбора по возрастанию.

После нажатия на “Mix” выполнится осуществляемая также случайным образом перестановка элементов рандомного массива: элементы остаются теми же, но встают на другие места.

После нажатия на “Descargar” (исп. “Скачать”) происходит загрузка массива из текстового файла. С ним впоследствии можно работать таким же образом, будто он был создан нажатием на “Crear”. При этом при нажатии на “Descargar” и “Crear” очищаются (если были заполнены) столбцы под панелями “Secuencia” и “Sum.

При этом если, ещё не создав случайный массив (не нажав на Button “Crear”), пробовать нажимать на другие кнопки (кроме “Descargar”), то на экране появится сообщение “Необходимо создать массив”. Так как мне кажется нелогичным пытаться воспроизводить действия с массивом, которого ещё не существует. Чтобы эти действия стали возможными, необходимо нажать либо на “Crear”, либо на “Descargar”.

Кроме того, возможно изменение значения любого элемента случайного массива и продолжение работы уже с новым значением. Также присутствует изображение, содержащее мои фамилию, имя и отчество, вариант и группу.

**Пример работы приложения**

Min = 1;  
Max = 45;  
Cantidad (исп. “Количество”) = 6;  
k = 4;

Crear->   
a[] = {1,5,10,1,5,2}. Появился текстовый файл, содержащий абсолютно те же значения, записанные через запятую.  
Содержание текстового файла - 1,5,10,1,5,2

Formula->  
x = 0,1,3,6,10,15.

Secuencia->  
b[] = {1,5,10,5}. Создаётся ещё один текстовый файл, в котором появляется эта последовательность.   
Содержание текстового файла – 1,5,10,5,,  
  
Suma->  
max == 21.

Sort->  
a[] = {1,1,2,5,5,10}.

Mix->  
a[] = {10,1,1,5,5,2}.

Descargar->  
a[] = {1,5,10,1,5,2}.

**Листинг кода**

**Unit1.cpp**

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

typedef struct ar{

int num;

int val;

} ar;

int k, n, c, max, pro, q; bool abcd;

ar a[30], p[30];

int itmp[30], valsum[30];

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

Image1->Picture->LoadFromFile("\image.bmp" );

abcd = false;

}

//---------------------------------------------------------------------------

int \_\_fastcall TForm1::factorial (int n)

{

if (n <= 0)

return 1;

else

return n\*factorial(n-1);

}

void \_\_fastcall TForm1::cnk(int m,int l)

{

//m-сколько осталоась выбрать,l - номер элемента с которого начниаем

int i, j, q, y, x, z, sum;

if (m == 0){ //Здесь - обработка комбинации

//printf ("\n");

x = 0; y = 0; z = 0; sum = 0;

for (j = 0; j < k; j++){

sum += p[j].val;

itmp[j] = p[j].num;

while (x < sum){

x = y + z;

y = x;

z++;

}

//printf ("%d ",p[j].val);

}

//printf("Sum = %d", sum);

if (sum == x){

//printf("\t-->Совпадение!");

if(sum > max){

max = sum;

for (j = 0; j < k; j++){

valsum[j] = itmp[j];

}

}

}

}else

for (i=l;i<=n-m;i++){

p[k-m].val=a[i].val;

p[k-m].num=a[i].num;

cnk(m-1,i+1);

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

randomize();

pro = StrToInt(Edit2->Text);

q = StrToInt(Edit1->Text);

n = StrToInt(Edit3->Text);

k = StrToInt(Edit4->Text);

if (n > 30)

{

ShowMessage("[ESP] Es prohibido tener mas que 30 elementos debido a largo tiempo de procesamiento / [ENG] It's prohibited to have more than 30 elements due to a large time of processing");

}

else if (k > n)

{

ShowMessage("[ESP] Es imposible encontrar un resultado si k>n / [ENG] It's impossible to find a result if k>n");

}

else

{

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

{

a[i].val = random(pro - q + 1) + q;

a[i].num = i+1;

StringGrid1->RowCount = n;

StringGrid1->Cells[0][i] = a[i].val;

}

abcd = true;

TStringList \*list = new TStringList;

list->Add(StringGrid1->Cols[0]->DelimitedText);

list->SaveToFile("create.txt");

delete list;

for (int i = 1;i<StringGrid1->ColCount; i++)

StringGrid1->Cols[i]->Clear();

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

if (abcd == false)

{

ShowMessage("[ESP] Es necesitado crear una matriz / [ENG] It's needed to create an array");

}

else

{

int x,y,z;

x = 0; y = 0; z = 0;

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

{

x = y + z;

y = x;

z++;

StringGrid1->Cells[1][i] = x;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

if (abcd == false)

{

ShowMessage("[ESP] Es necesitado crear una matriz / [ENG] It's needed to create an array");

}

else

{

max = 0;

cnk(k,0);

if(max == 0)

ShowMessage("[ESP] No hay convergencias / [ENG] There's no convergences");

else

{

StringGrid1->Cells[3][0] = max;

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

{

StringGrid1->Cells[2][i] = a[valsum[i]-1].val;

}

TStringList \*list = new TStringList;

list->Add(StringGrid1->Cols[2]->DelimitedText);

list->SaveToFile("sequence.txt");

delete list;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

if (abcd == false)

{

ShowMessage("[ESP] Es necesitado crear una matriz / [ENG] It's needed to create an array");

}

else

{

srand(time(NULL));

for (int i = n - 1; i >= 1; i--)

{

int j = rand() % (i + 1);

int tmp = a[j].val;

//int num = a[j].num;

a[j].val = a[i].val;

//a[j].num = a[i].num;

a[i].val = tmp;

//a[i].num = num;

}

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

{

StringGrid1->Cells[0][i] = a[i].val;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

if (abcd == false)

{

ShowMessage("[ESP] Es necesitado crear una matriz / [ENG] It's needed to create an array");

}

else

{

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

{

int minPosition;

minPosition = i;

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

if (a[minPosition].val > a[j].val)

minPosition = j;

int tmp = a[minPosition].val;

//int num = a[minPosition].num;

a[minPosition].val = a[i].val;

//a[minPosition].num = a[i].num;

a[i].val = tmp;

//a[i].num = num;

}

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

{

StringGrid1->Cells[0][i] = a[i].val;

}

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

int num;

TStringList \*list = new TStringList;

list->LoadFromFile("create.txt");

for (int i = 0;i<StringGrid1->ColCount; i++)

StringGrid1->Cols[i]->Clear();

for(int i = 0; i<list->Count; i++){

StringGrid1->Cols[i]->DelimitedText = list->Strings[i];

}

delete list;

num = StringGrid1->RowCount;

n = 0;

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

if(StringGrid1->Cells[0][i].Length() > 0) n++;

}

for(int i=0 ;i<30;i++){

a[i].val =0;

a[i].num = i+1;

}

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

{

try{

a[i].val = StrToInt(StringGrid1->Cells[0][i]);

}catch(...){

ShowMessage ("Произошла ошибка. - "+StringGrid1->Cells[0][i]+ n);

}

}

abcd = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::StringGrid1SelectCell(TObject \*Sender, int ACol,

int ARow, bool &CanSelect)

{

StringGrid1->Options = StringGrid1->Options << goEditing;

CanSelect = ACol == 0;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::StringGrid1GetEditText(TObject \*Sender, int ACol,

int ARow, AnsiString &Value)

{

for (int i = 1;i<StringGrid1->ColCount; i++)

StringGrid1->Cols[i]->Clear();

n = StringGrid1->RowCount;

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

{

a[i].val = StrToInt(StringGrid1->Cells[0][i]);

}

}

//---------------------------------------------------------------------------

**Unit1.h**

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ExtCtrls.hpp>

#include <Grids.hpp>

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TStringGrid \*StringGrid1;

TButton \*Button2;

TButton \*Button3;

TEdit \*Edit1;

TEdit \*Edit2;

TEdit \*Edit3;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TButton \*Button4;

TButton \*Button5;

TButton \*Button6;

TButton \*Button1;

TEdit \*Edit4;

TLabel \*Label4;

TImage \*Image1;

TPanel \*Panel1;

int \_\_fastcall factorial (int n);

void \_\_fastcall cnk(int m,int l);

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

void \_\_fastcall Button3Click(TObject \*Sender);

void \_\_fastcall Button5Click(TObject \*Sender);

void \_\_fastcall Button4Click(TObject \*Sender);

void \_\_fastcall Button6Click(TObject \*Sender);

void \_\_fastcall StringGrid1SelectCell(TObject \*Sender, int ACol, int ARow,

bool &CanSelect);

void \_\_fastcall StringGrid1GetEditText(TObject \*Sender, int ACol, int ARow,

AnsiString &Value);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif