МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №3

з дисципліни «Дискретна математика»

на тему: *«Комбінаторика: перестановки, розміщення, сполучення»*

Виконав:

студент гр.ІО-31

Долинний О.В.

Перевірив:

Новотарський М.А.

Київ 2014

**Лабораторна робота №3**

**Тема:** «Комбінаторика: перестановки, розміщення, сполучення».

**Мета:** вивчення правил утворення комбінацій можин: перестановок, розміщень, сполучень..

**Завдання:** вивчити алгоритми формування перестановок, сполучень та

розбиття; написати програми для виконання даних алгоритмів.

1. ***Завдання до лабораторної роботи***

Необхідно в рамках форми OperForm реалізувати алгоритм відповідно до варіанту лабораторної роботи.

Номер варіанту I визначається як результат операції I = NZK mod 14+1, де NZK – номер залікової книжки. Номер варіанту відповідає номеру пункту завдання до лабораторної роботи.

При виконанні завдань лабораторної роботи використати такий варіант:

**І= 3110 mod 14 + 1 = 3**

***Варіант 3:*** Виконати завдання 2 до лабораторної роботи за умови, що вхідні параметри приймають такі значення:

1. Максимальне значення *n* дорівнює номеру залікової книжки (NZK).

2. Значення *m* може змінюватися довільно від 1 до *n*

3. Сформувати початкову перестановку *b**n*,...,*b*0таким чином, що кожен її елемент вибирається випадково.

***Завдання 2:*** Вивчити принципи роботи алгоритму генерації двійкових векторів довжини *n* у лексикографічному порядку. Написати програму генерації двійкових векторів довжини *n*.

1. ***Теоретичні основи***

**Перестановки**

*Комбінації з n елементів, які відрізняються одна від одної тільки порядком елементів, називаються перестановками.*

Перестановки позначаються символом *Рn*, де *n* — число елементів, що входять у кожну перестановку.

*Добуток всіх натуральних чисел від 1 до n включно називають n-факторіалом* і пишуть:  *.* Вважають, що *0! = 1* і *n**N* . Основна властивість факторіала: *(n + 1)! = (n +1) ・ n!*. Отже, число перестановок обчислюємо за формулою: *Рn = n*!

**Розміщення**

*Комбінації з n елементів по m елементів, які відрізняються одна від одної або самими елементами, або порядком елементів, називаються розміщеннями.*

Розміщення позначаються символом  , де *n* – число всіх наявних елементів, *m –* число елементів у кожній комбінації. Число розміщень можна обчислити за формулою:

 *n**n* 1*n* 2*...**n* *m* 1*n* **,** де *0 ≤ m ≤ n; m, n**N.* Вважають, що  

Формулу розміщення можна записати у факторіальній формі:

Основні властивості розміщень:

1) 

2) 

**Сполучення**

*Сполученнями називаються всі можливі комбінації з n елементів по m, які відрізняються одна від одної принаймні хоча б одним елементом ( m, n**N і n* *m).*

У загальному випадку число сполучень із *n* елементів по *m* дорівнює числу розміщень з *n* елементів по *m*, діленому на число перестановок з *m* елементів: *.* Використовуючи для кількості розміщень і перестановок факторіальні формули  і   *,* одержимо формулу кількості сполучень у вигляді: 

Основні властивості сполучень:  ; 

**Визничення лексикографічного порядку.**

Нехай існують перестановки у вигляді послідовностей {x1,x2,x3,…,xn}, {y1,y2,y3,…,yn} однієї і тієї ж множини *X* Говорять, що перестановки з елементів множини *X* упорядковані у лексикографічному порядку , якщо {x1,x2,x3,…,xn}< {y1,y2,y3,…,yn} тоді і тільки тоді, коли для деякого  *xk**yk і xi=yi* для всіх *i* *k* .

**Визничення антилексикографічного порядку.**

Нехай існують перестановки у вигляді послідовностей {x1,x2,x3,…,xn}, {y1,y2,y3,…,yn} однієї і тієї ж множини *X* . Говорять, що перестановки з елементів множини *X* упорядковані у антилексикографічному порядку , якщо {x1,x2,x3,…,xn}> {y1,y2,y3,…,yn} тоді і тільки тоді, коли для деякого *k xk**yk і xi=yi* для всіх *i* *k* .

1. ***Роздруківка тексту програми***

Type

TArray = Array [0..100] of Integer;

procedure Download\_VectClick(Sender: TObject);

procedure ExitButtonClick(Sender: TObject);

procedure First\_VectorClick(Sender: TObject);

procedure GenerateClick(Sender: TObject);

procedure Input\_m\_nClick(Sender: TObject);

procedure Input\_VectClick(Sender: TObject);

var

m,n : integer;

b:TArray;

procedure TOperForm.Input\_m\_nClick(Sender: TObject);

var

i:integer;

begin

if Vector\_Input.Checked // input m,n from display

then

begin

for i:=1 to n do

First\_Vector.Cells[n-i,0]:='';

m:=StrToInt(Qual\_Vect\_Inp.text); // quality of vectors

n:=StrToInt(Qual\_Elem\_Inp.text); // quality of elements

for i:=1 to n do

First\_Vector.Cells[n-i,0]:='0';

end;

end;

procedure TOperForm.Input\_VectClick(Sender: TObject);

var

i:integer;

begin // input first vector from display

if Vector\_Input.Checked

then

for i:=0 to n-1 do

b[i]:=StrToInt(First\_Vector.Cells[n-i-1,0]);

end;

procedure TOperForm.First\_VectorClick(Sender: TObject);

begin

if First\_Vector.Cells[First\_Vector.Col,First\_Vector.Row]='0'

then First\_Vector.Cells[First\_Vector.Col,First\_Vector.Row]:='1

else First\_Vector.Cells[First\_Vector.Col,First\_Vector.Row]:='0';

end;

procedure TOperForm.Download\_VectClick(Sender: TObject);

var

i:integer;

str:string;

f:textfile;

begin // input first vector from file

if Vector\_File.Checked

then

begin

assignFile(f, 'DATA\P3.txt');

reset(f);

Readln(f, Str);

m:=StrToInt(Str); // quality of vectors

Qual\_Vect\_Inp.text:=IntToStr(m);

Readln(f, Str);

n:=StrToInt(Str); // quality of elements

Qual\_Elem\_Inp.text:=IntToStr(n);

i:=0;

while not (EOF(f)) do

begin

Readln(f, Str);

b[n-i-1]:=StrToInt(Str);

Inc(i);

end;

closefile(f);

for i:=0 to n-1 do

First\_Vector.Cells[n-i-1,0]:=IntToStr(b[i]);

end;

end;

procedure TOperForm.GenerateClick(Sender: TObject);

var i,m0:integer;

begin

if leksi.checked

then

begin

m0:=m; // quality of vectors

for i:=0 to m-1 do // write numbers of vectors

Result\_Vector.Cells[0,i]:=IntToStr(i+1);

b[n]:=0; // first condition

while (b[n]=0) do

begin

for i:=0 to n-1 do // output current vector

Result\_Vector.Cells[i+1,m0-m]:=IntToStr(b[n-i-1]);

m:=m-1; // rest of vectors

if (m=0) // check rest of vectors

then break

else

begin

i:=0; // find first position with 0

while (b[i]=1)and (i<m0-1) do

begin

b[i]:=0;

inc(i);

end;

b[i]:=1;

end;

end;

end;

if anti\_leksi.checked

then

begin

m0:=m; // quality of vectors

for i:=0 to m-1 do // write numbers of vectors

Result\_Vector.Cells[0,i]:=IntToStr(i+1);

b[n]:=1; // first condition

while (b[n]=1) do

begin

for i:=0 to n-1 do // output current vector

Result\_Vector.Cells[i+1,m0-m]:=IntToStr(b[n-i-1]);

m:=m-1; // rest of vectors

if (m=0) // check rest of vectors

then break

else

begin

i:=0; // find first position with 1

while (b[i]=0)and (i<m0-1) do

begin

b[i]:=1;

inc(i);

end;

b[i]:=0;

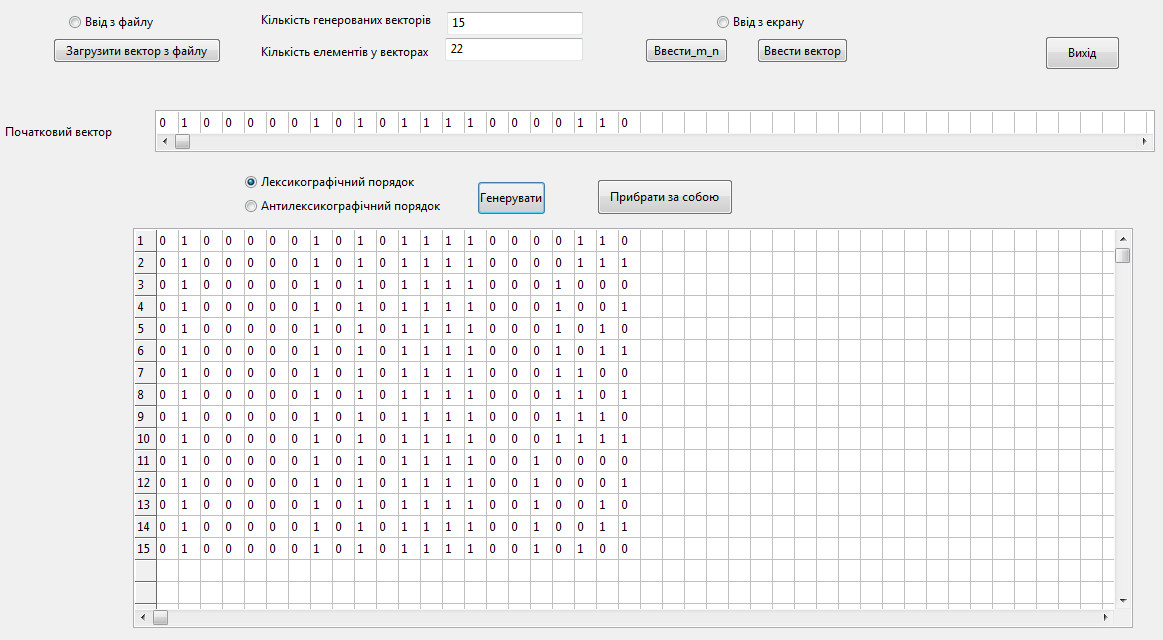
end;

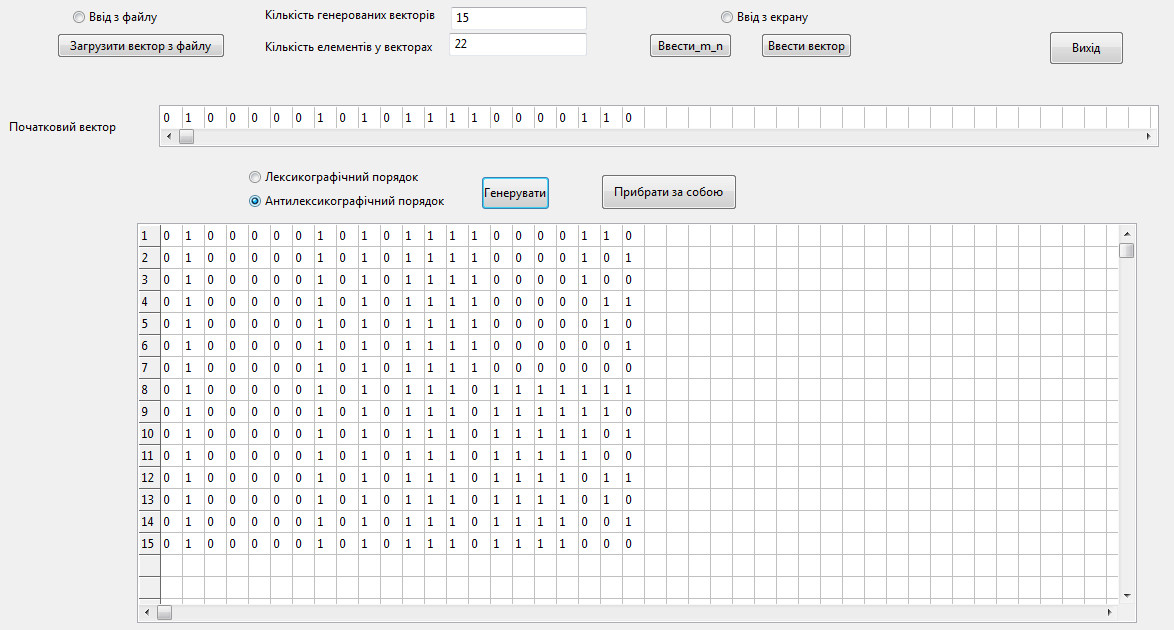
end;

end;

end;

1. ***Роздруківка результатів виконання програми***

******

******

1. ***Висновки***

*У ході лабораторної роботи я закріпив знання з комбінаторики, навчився використовувати набуті знання на практиці. При виконанні роботи були сгенеровані двійкові вектори у лексикографічному і анти лексикографічному порядку.*

*Отримані результати підтверджують правильність роботи алгоритму.*

*Під час виконання лабораторної роботи я покращив навики роботи за програмами з візуальними частинами.*