**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Дискретна математика**

**Лабораторна робота №5**

**«Комбінаторика: перестановки, розміщення, сполучення»** Виконав:

студент групи IО-63

Братун Андрій

Залікова книжка № 6305

Перевірив Новотарський М. А.

Київ 2016 р.

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

**Загальне завдання:** Вивчити алгоритми формування перестановок, сполучень та розбиття. Написати програми для виконання даних алгоритмів.

**Основні означення :**

Перестановки Комбінації з n елементів, які відрізняються одна від одної тільки порядком елементів, називають перестановками. Перестановки позначають символом Рn, де n — число елементів, що входять у кожну перестановку. Приклад. Нехай множина М містить три букви А, В, С. Складемо всі можливі комбінації із цих букв: АВС, АСВ, ВСА, CAB, CBA, ВАС (усього 6 комбінацій). Видно, що вони відрізняються одна від одної тільки порядком розташування букв. Добуток всіх натуральних чисел від 1 до n включно називають n-факторіалом і пишуть: n!= 1 · 2 · 3 · ... · (n - 1) · n. Вважають, що 0! = 1 і nN  . 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, називають розміщеннями. Розміщення позначають символом m nА , де n – число всіх наявних елементів, m – число елементів у кожній комбінації. Число розміщень можна обчислити за формулою:      211  nmn...nnА m n , де 0 ≤ m ≤ n; m, nN.

Вважають, що 1 А0 n  .

90

Приклад. Нехай множина M містить чотири букви А, В, С, D. Склавши всі комбінації тільки із двох букв, одержимо: АВ, AC, AD, ВА, ВР, BD, СА, СВ, CD, DA, DB, DC. Формулу розміщення можна записати у факторіальній формі:

 m!n n!

Аm n   . Основні властивості розміщень: 1)   nmAА m n m1 n   ;

2) n! PА n n n  .

Сполучення Сполученнями називають 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:

m

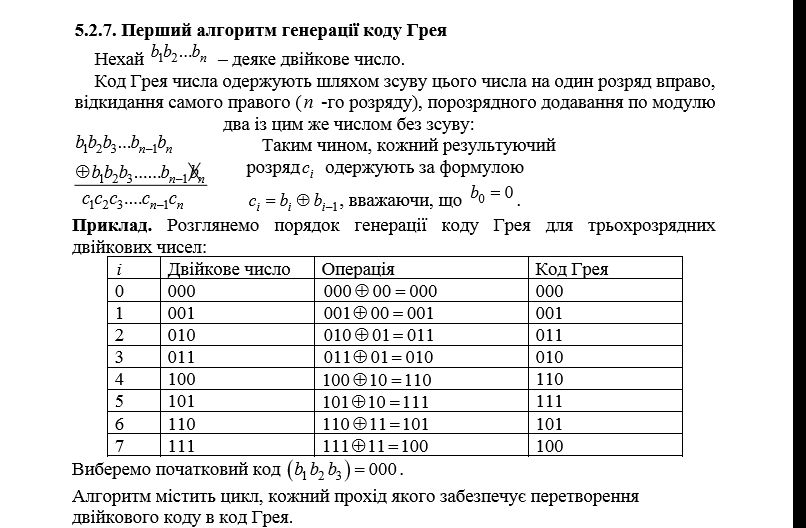
m nm n P A С  . Використовуючи для кількості розміщень і перестановок факторіальні формули  n!m m! Аm n   і n! Pn  , одержимо формулу кількості сполучень у вигляді:  m!m!n n! Cm n   . Основні властивості сполучень:  m!m!n n! PP P C mmn nnm n       ; nm n m n CC   . Приклад. Множина М утворена з чотирьох букв А, В, С, D. Скласти комбінації з двох букв, що відрізняються хоча б одним елементом. Маємо АВ, AC, AD, ВР, BD, CD. Виходить, що кількість сполучень з

чотирьох елементів по два дорівнює 6. Це коротко записують так: 6 С2 4  .

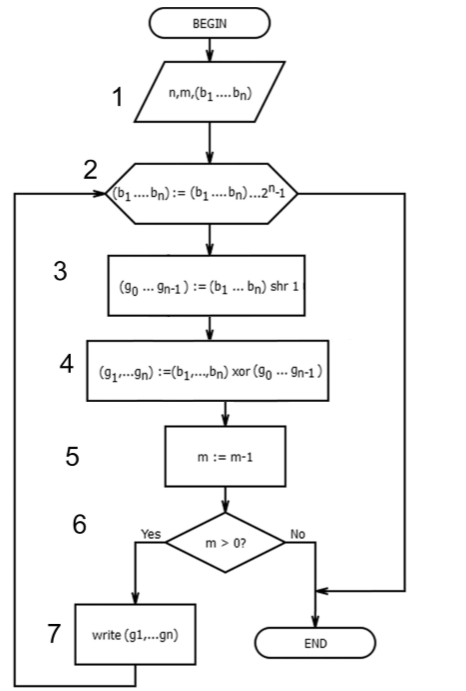
Розміщення з повтореннями Розміщення з n елементів по k відображають упорядковані комбінації різних елементів множини М, |М|=n. Часто доводиться утворювати упорядковані комбінації з повтореннями деяких елементів. Наприклад, з множини М = {A, Б}

91

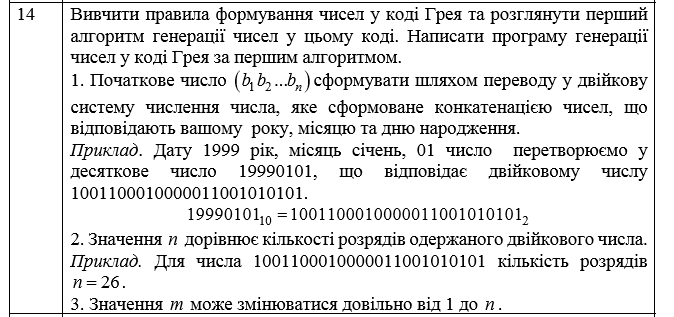
можна утворити вісім комбінацій з трьох елементів: ААА, ААБ, АБА, БАА, БАБ, ББА, АББ, БББ. Тут n = 2, k = 3. Такі упорядковані k-комбінації називають кортежами довжини k. Два кортежі (тобто дві загальні комбінації) вважають однаковими, якщо вони мають однакову довжину і на місцях з однаковими номерами стоять однакові елементи. Кортеж довжини k з n елементів називають розміщенням з повтореннями з n елементів по k . Кількість кортежів обчислюють за формулою: kk nА n  . Дійсно, після заповнення першого місця кортежу довжиною k одним з n елементів (що можливо зробити n варіантами) заповнити друге місце кортежу можна знову будь-яким елементом з усієї множини (повторюючи в одному з варіантів елемент, який знаходиться на першому місці), і так далі k разів. За правилом добутку одержимо, що kk n k разів А n n n



**Блок схема:**

****

**Варіант виразу відповідно до індивідуального завдання – 14**

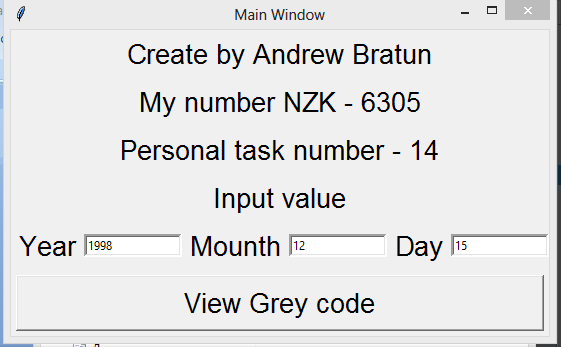
****

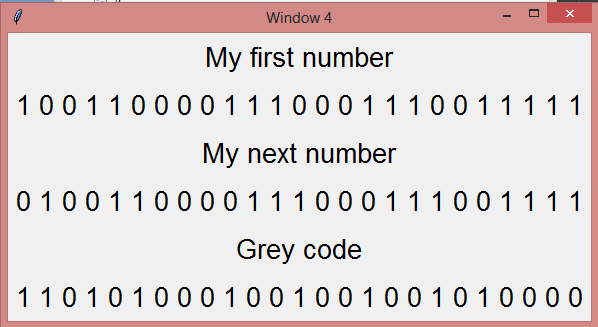
**Роздруківка того фрагменту тексту програми, який написаний індивідуально.**

**Module Main**

**from** tkinter **import** \*  
**from** tkinter **import** messagebox  
  
root = Tk()  
root.title("Main Window")  
  
#root.maxsize(width=320, height=190)  
#root.minsize(width=320, height=190)  
  
  
# My uniq task  
NZK = 6305  
number = NZK % 26 + 1  
task = "Personal task number - " + str(number)  
  
  
# #####################MAINLOOOOOOOP################  
  
**def get\_number**():  
 input\_number = ent1.get() + ent2.get() + ent3.get()  
 int\_number = int(input\_number)  
 bin\_number = bin(int\_number)  
 **global** str\_result  
 **global** next\_number  
 str\_bin\_number = str(bin\_number)  
 str\_result = str\_bin\_number[2:]  
 n = len(str\_result)  
 m = 20  
 #m 1 to n  
 next\_number = '0' + str\_result[:-1]  
 zip\_list = list(zip(str(next\_number),str\_result))  
 **global** result  
 result = []  
 **for** i **in** range(len(zip\_list)):  
 **if** zip\_list[i][0] == zip\_list[i][1]:  
 result.append(0)  
 **else**:  
 result.append(1)  
  
  
  
**def callback**(result,str\_result,next\_number):  
 window1 = Toplevel()  
 window1.title("Window 4")  
 text1 = result  
 text2 = []  
 **for** i **in** list(str\_result):  
 text2.append(int(i))  
 text3 = []  
 **for** i **in** list(next\_number):  
 text3.append(int(i))  
 lab1 = Label(window1, text=text2, font='arial 20')  
 lab2 = Label(window1, text=text3, font='arial 20')  
 lab3 = Label(window1, text=text1, font='arial 20')  
 lab4 = Label(window1, text="My first number", font='arial 20')  
 lab5 = Label(window1, text="My next number", font='arial 20')  
 lab6 = Label(window1, text="Grey code", font='arial 20')  
  
 lab1.grid(row=2, column=1, sticky=W + E + N + S, pady=5, padx=5)  
 lab2.grid(row=4, column=1, sticky=W + E + N + S, pady=5, padx=5)  
 lab3.grid(row=6, column=1, sticky=W + E + N + S, pady=5, padx=5)  
 lab4.grid(row=1, column=1, sticky=W + E + N + S, pady=5, padx=5)  
 lab5.grid(row=3, column=1, sticky=W + E + N + S, pady=5, padx=5)  
 lab6.grid(row=5, column=1, sticky=W + E + N + S, pady=5, padx=5)  
  
**def knopka**(event):  
 get\_number()  
 callback(result,str\_result,next\_number)  
  
  
  
# Info about student  
lab1 = Label(root, text='Create by Andrew Bratun', font='arial 20')  
lab2 = Label(root, text='My number NZK - 6305', font='arial 20')  
lab4 = Label(root, text=task, font='arial 20')  
  
lab5 = Label(root, text="Input value", font='arial 20')  
lab6 = Label(root, text="Year", font='arial 20')  
lab7 = Label(root, text="Mounth", font='arial 20')  
lab8 = Label(root, text="Day", font='arial 20')  
  
ent1 = Entry(root,width=15, bd=3)  
ent2 = Entry(root,width=15, bd=3)  
ent3 = Entry(root,width=15, bd=3)  
  
but1 = Button(root, text='View Grey code', font='arial 20')  
but1.bind("<Button-1>", knopka)  
  
  
# Info about student  
lab1.grid(row=1, column=1, columnspan=6, sticky=W+E+N+S, pady=5, padx=5)  
lab2.grid(row=2, column=1, columnspan=6, sticky=W+E+N+S, pady=5, padx=5)  
lab4.grid(row=3, column=1, columnspan=6,sticky=W+E+N+S, pady=5, padx=5)  
  
lab5.grid(row=4, column=1, columnspan=6,sticky=W+E+N+S, pady=5, padx=5)  
lab6.grid(row=5, column=1,sticky=W+E+N+S, pady=5, padx=5)  
lab7.grid(row=5, column=3, sticky=W+E+N+S, pady=5, padx=5)  
lab8.grid(row=5, column=5, sticky=W+E+N+S, pady=5, padx=5)  
  
  
  
ent1.grid(row=5,column=2)  
ent2.grid(row=5,column=4)  
ent3.grid(row=5,column=6)  
  
  
but1.grid(row=6, column=1, columnspan=6,sticky=W+E+N+S, pady=5, padx=5)  
  
  
root.mainloop()

**Роздруківка результатів виконання програми з контрольним прикладом**

****

****

**Аналіз результатів та висновки**

Протягом виконання роботи я вивчив основні методи роботи з графами, вивчив їх види та алгоритми пошуку шляхів, закріпив занання GUI в модулі tkinter в Python.

Складним завдання виявилося від лагодження програми на етапі розробки, оскільки виникпало багато логічних помилок які найскладніше виявити, проте завдання було цікавим, розробка своєї програми з інтерфейсом мені сподобалася. В ході розробки програми виникали деякі питання, але все вдалося знайти за допомогою додаткової інформації в інтернеті.