Міністерство освіти і науки України

Львівський національний університет

імені Івана Франка

Кафедра

обчислювальної математики

Янчинський Юрій Владленович

Звіт з обчислювальної практики

(Python, C++)

Студент групи ПМп-11

Керівник

Гарасим Ярослав Степанович

Львів-2016

Зміст

Вступ…………………………………………..………………………………………......................3

Завдання 1…………………………………………………………………………......................4

Завдання 2…………………………………………………………………………......................5

Завдання 3……………………………………………………………………………....................6

Завдання 4…………………………………………………………………………......................7

Завдання 5…………………………………………………………………………......................8

Завдання 6…………………………………………………………………………....................10

Завдання 7…………………………………………………………………………....................11

Завдання 8…………………………………………………………………………....................12

Завдання 9…………………………………………………………………………....................15

Завдання 10……………………………….………………………………………....................17

Завдання 11………………………………………………………….……………....................19

Висновок……………………………………………………………………………....................20

Список літератури……………………………………………………………......................21

Вступ

Програмування – це процес створення алгоритму, написання його згідно з вимогами певної мови програмування, тестування (але не оптимізування) і отримання зарплати за якісне виконання вище згаданих пунктів. Воно поєднує в собі елементи інженерії , фундаментальних наук та мистецтва, а обмеженням є лише наша уява.

На першому курсі ми вивчали (і деякі з нас вивчили) такі мови програмування як Python та C++ .

Python – одна з найперспективніших мов програмування на сьогодні, програмісти всього світу визнають її великий потенціал. Багато складних операцій можна виконати кількома рядками , що говорить про простоту і означає , що ця мова добре підходить початківцям. С++ – це мова, що базується на досить старій мові С (тоді як Python – просто написаний на ній) . Тут потрібно вручну реалізовувати чимало того , що на Python робилося в один рядок (або просто підключити потрібні бібліотеки С++) . Програмування на цій мові дає розуміння алгоритмів і підходить тим людям, які опанували мови типу Pascal, але розчарує прихильників мови Python. Лише добре орієнтуючись в алгоритмах ,можна ефективно програмувати на С++ .

Завдання 1

Постановка задачі

Для заданого n обчислити суму

(1+2+3+…+n)2

Опис алгоритму

Потрібно зчитати з клавіатури кількість елементів (n);

Описати цикл від 1 до n;

В циклі до суми елементів, яка спочатку була рівна нулю, додавати поточний елемент ;

Вивести на екран значення суми у квадраті.

Текст програми (Python)

a=0

n=int(input('n='))

for i in range(1,n+1):

a=a+i

print(a\*\*2)

i=int(input())

Апробація

Вхідні дані:

Кількість елементів (n)=5;

Вихідні данні:

Сума елементів у квадраті=225;

Коментарі:

Рядок «i=int(input())» запобігає закриттю командної стрічки після виведення відповіді.

Завдання 2

Постановка задачі

Прочитати n (n<=9) цілих чисел. Знайти кількість чисел, які містять цифру 5. Масиви та стрічки не використовувати.

Опис алгоритму

Зчитуємо кількість чисел (n);

Описати цикл від 1 до n;

В циклі задаємо ще 1 цикл «while», в якому перевіряємо, чи остання цифра числа рівна 5, і відкидаємо цю цифру доти, доки доки вона не буде рівна 5(тоді додаємо до кількості шуканих чисел 1) або число не стане нулем (тоді, не додаємо), і переходимо до наступного числа;Виводимо на екран кількість шуканих чисел;

Текст програми (Python)

n=int(input('n='))

m=0

for i in range (n):

a=abs(int(input('a=')))

z=1

while a>0 and z==1:

if a%10==5:

m=m+1

z=0

a=a//10

print(m)

i=int(input())

Апробація

Вхідні дані:

Кількість чисел (n)=5;

Число =1234

Число =1235

Число =1255

Число =5678

Число =5555

Вихідні данні:

Кількість шуканих чисел=4;

Коментарі:

Змінна «m» - лічильник;

Змінна «z» -умова виходу з циклу, якщо число містить цифру5;

Завдання 3

Постановка задачі

Прочитати дійсну матрицю розміром m\*n (m,n>5). Надрукувати індекси сідлових точок .

Опис алгоритму

Зчитуємо розміри матриці (m,n);

Зчитуэмо матрицю;

Для кожного елемента матриці перевіряємо, чи є він одночасно найбільшим в рядку і найменшим в стовпчику, якщо так, то виводимо його індекси;

Для кожного елемента матриці перевіряємо, чи є він одночасно найменшим в стовпчику і найбільшим в рядку, якщо так, то виводимо його індекси;

Текст програми (Python)

m=int(input('m='))

n=int(input('n='))

a=[]

for i in range(m):

a.append([int(input())for k in range(n)])

for i in range(len(a)):

print(a[i])

for i in range(m):

for j in range(n):

if a[i][j]==max(a[i][k] for k in range(n)):

if a[i][j]==min(a[k][j] for k in range(m)):

print(i+1,j+1)

for i in range(m):

for j in range(n):

if a[i][j]==min(a[i][k] for k in range(n)):

if a[i][j]==max(a[k][j] for k in range(m)):

print(i+1,j+1)

Апробація

Вхідні дані:

Висота матриці (m)=3;

Ширина матриці (n)=2;

Елементи матриці = 1,2,3,4,5,6;

Вихідні дані:

Матриця має вигляд

[1, 2, 3]

[4, 5, 6]

Індекси сідлових елементів = 1 3 і 2 1;

Завдання 4

Постановка задачі

Прочитати текстову стрічку довжиною не довше ніж 253 символи. Замінити кожне входження словосполучення «x» на «iks».

Опис алгоритму

Зчитуємо текстову стрічку;

За допомогою вбудованої функції заміняємо «x»на«iks»;

Виводимо результат;

Текст програми (Python)

s=input('s=')

a=s.replace('x','iks')

s=a

print(s)

Апробація

Вхідні дані:

Текстова стрічка (s) = xyzxxyzxxx;

Вихідні дані:

Результат = iksyziksiksyziksiksiks;

Завдання 5

Постановка задачі

Ввести n мірний масив трицифрових цілих додатніх чисел. Всі числа, які діляться на 3 або закінчуються цифрою 3 замінити на знакове «-3». Виконати циклічний зсув у масиві на k елементів та роздрукувати масив.

Опис алгоритму

Зчитуємо розмір масиву (n);

Зчитуємо відстань зсуву (k);

Зчитуємо елементи масиву;

Якщо остача від ділення елемента на 3 = 0 або його остання цифра 3, то заміняємо його на -3;

Створюємо порожній масив b;

Зсув вправо (k>0): в циклі від 1 до n, якщо номер <k, то до масиву b додається елемент № i+n-k з початкового масиву, якщо номер>k, то до масиву bдодається елемент № i-k;

Зсув вліво (k<0) : в циклі від 1 до n, якщо номер + |k|<n, то до масиву b додається елемент № i+|k|з початкового масиву, якщо номер + |k| >n, до масиву b додається елемент № i+|k|-n;

Виводимо отриманий масив.

Текст програми (Python)

n=int(input('n='))

k=int(input('k='))

a=[]

b=[]

for i in range(n):

a.append(int(input()))

print(a)

for i in range(n):

if a[i]%3==0 or a[i]%10==3:

a[i]=-3

print(a)

if k>0:

for i in range(n):

if i<k:

b.append(a[i+n-k])

else:

b.append(a[i-k])

else:

k=abs(k)

for i in range(n):

if i+k<n:

b.append(a[i+k])

else:

b.append(a[i+k-n])

print(b)

Апробація

Вхідні дані:

розмір масиву (n) = 5;

відстань зсуву (k) = 3;

елементи масиву = 1,3,32,23,6;

Вихідні дані:

Масив після заміни має вигляд[1, -3, 32, -3, -3];

Масив після зсуву має вигляд[32, -3, -3, 1, -3].

Завдання6

Постановка задачі

Описати рекурсивну функцію, яка для заданого n (n>2) обчислює значення виразу :

Опис алгоритму

Описуємо рекурентну функцію, яка приймає поточне значення кореня суми доданків (s) і значення наступного доданку (y);

Функція обчислює корінь суми поточної суми і наступного доданка(сумувати починаємо з кінця, тобто з 2n+1 ) і зменшує значення наступного доданка на 2;

Якщо значення наступного доданка досі > 0, то рекурсивно викликаємо функцію для поточної суми і наступного доданка,в іншому випадку виводимо на екран значення поточної суми;

В основній програмі зчитуємо з клавіатури n і викликаємо функцію для поточної суми 0 і наступногододанка2n+1.

Текст програми (Python)

def next(s,y):

import math

s=math.sqrt(s+y)

y=y-2

if y>-1:next(s,y)

else: print(s)

n=int(input('n='))

next(0,2\*n+1)

Апробація

Вхідні дані:

n=5;

Вихідні дані:

≈ 1.85

Завдання7

Постановка задачі

Нехай задано натуральне число n , знайти суму цифр в записі цього числа.

Опис алгоритму

Зчитуємо число;

В циклі до суми цифр (c) додаємо остачу від ділення числа на 10 і присвоюємо числу значення цілої частини від ділення цього числа на 10 поки число не стане 0;

Виводимо суму цифр;

Текст програми (C++)

#include "iostream"

using namespace std;

int main()

{

Int n,c;

cin>> b;

c = 0;

while (n> 0)

{

c = c+b % 10;

n = n / 10;

}

cout<< c;

cin>> n;

}

Апробація

Вхідні дані:

Число n = 12345;

Вихідні дані:

Сума цифр = 15.

Завдання 8

Постановка задачі

Ввести n мірний масив трицифрових цілих додатніх чисел. Всі числа, які діляться на 3 або закінчуються цифрою 3 замінити на знакове «-3». Виконати циклічний зсув у масиві на k елементів та роздрукувати масив. Код підпрограм для заміни, зсуву та виводу розмістити в окремому\*.cpp файлі, декларацію підпрограм оформити у вигляді \*.h файла. Код основної програми повинен містити лише виклики підпрграм.

Опис алгоритму

Функції:

Функція отримання кількості елементів (getamount.cpp ):

Зчитуємо розмір масиву (n).

Функція зчитування масиву (createmass.cpp):

Зчитуємо елементи масиву.

Функція заміни (replace.cpp):

Якщо остача від ділення елемента на 3 = 0 або його остання цифра 3, то заміняємо його на -3.

Функція зсуву(moove.cpp):

Зчитуємо відстань зсуву (k);

Створюємо порожній масив b;

Зсув вправо (k>0): в циклі від 1 до n, якщо номер <k, то до масиву bдодається елемент № i+n-k з початкового масиву, якщо номер>k, то до масиву b додається елемент № i-k;

Зсув вліво (k<0) : в циклі від 1 до n, якщо номер + |k|<n, то до масиву bдодається елемент № i+|k|з початкового масиву, якщо номер + |k| >n, до масиву b додається елемент № i+|k|-n;

Функція виводу масиву(printmass.cpp):

Виводимо отриманий масив.

Створюємо і підключаємо відповідні \*.h файли: getamount.h, createmass.h, replace.h, moove.h, printmass.h.

Основна програма:

Виклик getamount;

Виділення пам’яті для n мірних масивів a,b;

Виклик createmass, printmass, replace, printmass, moove, printmass.

Текст програми (C++)

replace.cpp

#include <iostream>

#include "replace.h"

using namespace std;

int\* replace(int \*a,int n)

{

int i;

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

{

if ((a[i] % 3 == 0) || (a[i] % 10 == 3))

{

a[i] = -3;

}

}

return(a);

}

moove.cpp

#include <iostream>

#include "moove.h"

using namespace std;

int\* moove( int \*a, int n)

{

Inti,k;

int \*b = new int[n];

cout<<"k=";

cin>>k;

k=k%5;

if (k>0)

{

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

{

if (i<k)

{b[i]=a[i+n-k];}

else

{b[i]=a[i-k];}

}

}

else

{

k=abs(k);

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

{

if (i+k<n)

{b[i]=a[i+k];}

else

{b[i]=a[i+k-n];}

}

}

return(b);

}

main.cpp

#include <iostream>

#include"replace.h"

#include"moove.h"

#include"getamount.h"

#include"createmass.h"

#include"printmass.h"

using namespace std;

int main()

{

int n,i,k;

n = getamount();

int \*a = new int[n];

int \*b = new int[n];

a = createmass(n);

printmass(a,n);

a = replace(a,n);

printmass(a,n);

b = moove(a,n);

printmass(b,n);

system("pause");

}

Апробація

Вхідні дані:

розмір масиву (n) = 5;

елементи масиву = 1,3,32,23,6;

відстань зсуву (k) = 3;

Вихідні дані:

Масив після заміни має вигляд[1, -3, 32, -3, -3];

Масив після зсуву має вигляд[32, -3, -3, 1, -3].

Завдання9

Постановка задачі

Написати підпрогорами (типу PUSH, POP) для роботи зі стеком на базі двонаправленого списку. Викоростовуючи створені підпрограми в основній програмі зчитати масив цілих чисел та вивести в оберненому порядку спочатку парні, а потім непарні числа.

Опис алгоритму

Створюємо структуру двонаправленого списку (Stack.cpp):

Кожен елемент списку має цифрове значення, вказівники на наступний і попередній елементи;

Описуємо функції додавання елемента в список, його виведення і очищення списку.

Описуємо підпрограми створення списку і виведення елементів у заданому порядку (createsort.cpp).

В основній програмі створюємо список, сортуємо і виводимо його, використовуючи описані функції (main.cpp).

Текст програми (C++)

Stack.cpp

structDoubleList

{

int data;

DoubleList \* next;

DoubleList \* prev;

};

DoubleList \*head = NULL;

DoubleList \*newCell = NULL;

DoubleList \*current = NULL;

DoubleList \*last = NULL;

void PUSH(int value)

{

if (head != NULL)

{

newCell = new DoubleList;

newCell->data = value;

newCell->next = NULL;

newCell->prev = current;

current->next = newCell;

last = newCell;

current = newCell;

}

else

{

newCell = new DoubleList;

newCell->data = value;

head = newCell;

newCell->next = NULL;

newCell->prev = NULL;

current = newCell;

last = head;

}

}

createsort.cpp

void createsort(int n)

{

Int i;

int\* a=new int[n];

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

{

cout<<"a"<<i<<"=";

cin>>a[i];}

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

{

if (a[n-1-i]%2==0)

{PUSH(a[n-1-i]);}

}

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

{

if (a[n-i-1]%2!=0)

{PUSH(a[n-i-1]);}

}

}

main.cpp

intmain()

{

int n,i;

cout<<"n=";

cin>>n;

createsort(n);

Show();

Destroy();

system("PAUSE");

return 0;

}

Апробація

Вхідні дані:

Кількість елементів списку () = 10;

Елементи списку = 1,2,3,4,5,6,7,8,9,0;

Вихідні дані:

Елементи у потрібному порядку: 0,8,6,4,2,9,7,5,3,1.

Завдання10

Постановка задачі

Написати клас, який моделює роботу з множиною. По двох заданих графах знайти їх різницю.

Опис алгоритму

Створюємо клас множин, в ньому описуємо функції для додавання елемента (з перевіркою на наявність його в множині )(add), виведення елементів (show), повернення елемента (give), повернення кількості елементів (givenumber), віднімання іншої множини (minus) і очищення множини (~setelem);

У головній програмі хардкодимо елементи двох графів, виводимо графи і їхню різницю за допомогою описаних функцій.

Текст програми (C++)

Classsetelem

{

private:

int\*a;

int n;

int k;

public:

setelem(int count=50)

{

k=0;

n=count;

a=newint[n];

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

{a[i]=0;}

};

Voidminus(setelemsub)

{

Intm=sub.givenumber(),x;

cout<<"result: ";

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

{

x=1;

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

{

if((a[i]==sub.give(j))||(sub.give(j)==(a[i]%10)\*10+a[i]/10))

{x=0;}

}

if(x)

{cout<<"("<<a[i]/10<<" "<<a[i]%10<<")"<<" ";}

}

cout<<"\n";

};

};

intmain()

{

setelem g1,g2;

g1.add(12);

g1.add(23);

g1.add(34);

g2.add(13);

g2.add(32);

g2.add(21);

g1.show();

g2.show();

g1.minus(g2);

system("pause");

}

Апробація

Коментарі:

Дуги задаються як двоцифрові числа (перша цифра – початок дуги, друга – її кінець).

Вхідні дані:

(Дуги графа хардкодяться в основій програмі)

Вихідні дані:

Дуги першого графа: (1, 2), (2, 3), (3, 4);

Дуги другого графа: (1, 3), (3, 2), (2, 1);

Різниця графів (перший – другий): (3, 4).

Завдання11

Постановка задачі

Зчитати текстову стрічку, видалити з неї повторні входження символів (не використовуючи бібліотеку stringта її фуекції).

Опис алгоритму

Виділяємо пам’ять під масив знаків довжиною 253;

Зчитуємо стрічку;

В циклі проходимо по всіх елементах масиву, якщо будь-який з наступних елементів (подвійний цикл) співпадає з поточним, то він замінюється на ‘~’;

В ще одному циклі проходимо по всіх елементах масиву, якщо поточний елемент не є ‘~’, то вінвиводиться на екран.

Текст програми (C++)

Intmain()

{

char s1[253];

int l,i,j;

cout<<"s1=";

gets(s1);

l=strlen(s1);

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

{

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

{

if (s1[i]==s1[j])

{s1[i]='~';}

}

}

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

{

if (s1[j]!='~')

{cout<<s1[j];}

}

cout<<"\n";

system("pause");

}

Апробація

Вхідні дані:

Текстова стрічка (s1)=afdsffdfafad;

Вихідні дані:

Текстова стрічка без повторень = afds;

Висновок

Програмування на Python показало нам всі кольори веселки і безмежні можливості сучасних комп’ютерів, як кілька рядків коду можуть розв’язати завдання , над яким людина працювала б щонайменше кілька годин . Але під час роботи в ній не варто надміру захоплюватися її можливостями, бо якщо перетворити сторінку коду в 1 рядок, то наврядче хтось крім вас зможе в ньому розібратися.

Це справді хороша мова програмування і вона ідеально підходить для початківців , бо не вимагає від програміста глибоких знань про принципи роботи комп’ютера.

Програмування на С++ на багато більш складне , але надає масу нових можливостей. Ця мова вимагає вже набагато глибших знань про внутрішні процеси , які відбуваються у процесорі . С++ краще підходить для роботи над великими проектами ніж Python принаймі за рахунок наявності \*.h файлів, які дозволяють розбити довгу програму на менші частини.

Список літератури

1. Brian W.Kernighan The С programming language (Second edition) /D.M.Ritchie/ PRENTICE HALL, Englewood Cliffs, New Jersie 07632
2. Peter van der Linden Expert С programming deep С secrets / PRENTICE HALL, Englewood Cliffs, New Jersie 07632
3. Zed A. Shaw Learning c the hard way / 200 Old Tappan Road, Old Tappan, New Jersie 07632