Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Кафедра обчислювальної математики

Кирильчук Володимир Володимирович

Звіт з навчальної практики

(Python, C++)

Керівник:

Гарасим Я.С.

Захищено з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів-2016

Зміст

Вступ 3

Завдання №1 4

Завдання №2 5

Завдання №3 8

Завдання №4 10

Завдання №5 11

Завдання №6 14

Завдання №7 15

Завдання №8 17

Завдання №9 19

Завдання №10 20

Завдання №11 22

Висновок 25

Cписок літератури 26

Вступ

Програмування – процес проектування, написання, тестування і підтримки комп’ютерних програм. Воно поєднує в собі елементи інженерії, фундаментальних наук та мистецтва.

На першому курсі ми вивчили такі мови програмування як Python і C++.

Python – одна з найперспективніших мов програмування на сьогодні, програмісти всього світу відзначають 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 – досить «молода» мова ). Тут потрібно вручну реалізовувати чимало того , що на Python робилося в один рядок (принаймні , якщо ви погано знаєте бібліотеки С++). Програмування на цій мові дає розуміння алгоритмів, лише добре орієнтуючись в алгоритмах , можна ефективно програмувати на С++. Також на цій мові написано більше половини усіх програм.

Завдання 1

Постановка задачі

Для заданого n обчислити суму - 2 + 3 – 4 +…+ (-1)n+1n.

Код програми(Python)

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

S=0

if n>0:

if n%2==0:

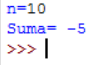
S=S-n//2

else:

S=S+(n+1)//2

print ('Suma=', S)

Апробація



Завдання 2

Постановка задачі

Для заданого n (n>5) надрукувати всю послідовність чисел Фібоначчі (f0=1,f1=1,fi=fi-1+fi-2,i=2,3,...).

Код програми(Python)

print 'Fibonacci numbers'

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

n0=1; n1=1

if 5<n<=100:

while n>0:

print n0

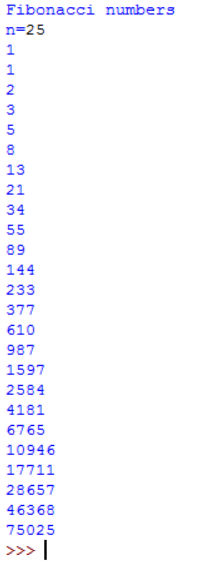
n2=n0+n1

n0=n1

n1=n2

n-=1

Апробація



Завдання 3

Постановка задачі

Прочитати дійсну матрицю розміром m×n (m,n >5). Впорядкувати стрічки матриці( сортування з вибором найбільшого) по неспаданню сум їх елементів.

Код програми(Python)

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

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

if m and n > 5:

Amt=[]

for i in range(m):

row=input('Type ' + str(n) + ' elements of row ' + str(i+1) + ':').split()

for i in range(len(row)):

row[i]=int(row[i])

Amt.append(row)

for i in range(m):

Amt[i].sort()

Amt[i].reverse()

Amt.sort()

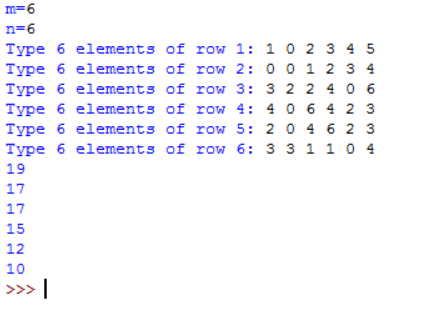
Amt.reverse()

for i in range(m):

Compofsums=sum(Amt[i])

print(Compofsums)

Апробація



Завдання 4

Постановка задачі

Прочитати текстову стрічку довжиною не більше ніж 253 символи. Вилучити зі стрічки всі пропуски, з яких розпочинається та якими закінчується стрічка та надрукувати її в оберненому порядку.

Код програми(Python)

a=input('text:')

if len(a)>253:

print ('Error')

else:

while a[0]==' ':

a=a[1:]

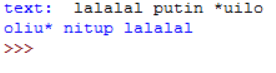
while a[-1]==' ':

a=a[:-1]

a=a[::-1]

print(a)

Апробація



Завдання 5

Постановка задачі

Прочитати m груп (2≤n≤10) до n в групі (4≤n≤50) цілих чисел. В кожній групі визначити, яких чисел більше: додатніх чи від’ємних та відповідно знайти суму цифр з більшої частини. Написати підпрограми (функції або/та процедури) для обчислень та перевірок вказаних властивостей, читання і друкування чисел.

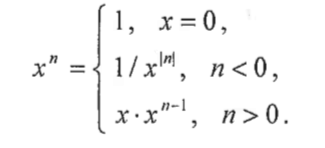
Код програми(Python)

**Апробація**

Завдання 6

Постановка задачі

Описати рекурсивну функцію pow(x,n) від цілого n та дійсного x(x≠0), яка обчислює xn по формулі



Код програми(Python)

def pow(x,n):

if x==0:

return 1

else:

if n<0:

return(1/x\*\*abs(n))

if n>0:

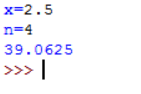
return(x\*x\*\*(n-1))

x=float(input('x='))

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

print(pow(x,n))

**Апробація**



Завдання 7

Постановка задачі

Нехай задано ціле число m. Отримати найбільше ціле число k, при якому 4k < m.

Код програми(C++)

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int m; double k;

cin >> m;

cout << "m=" << m << endl;

k = log10(m) / log10(4);

if (!(k - (int) k))

k--;

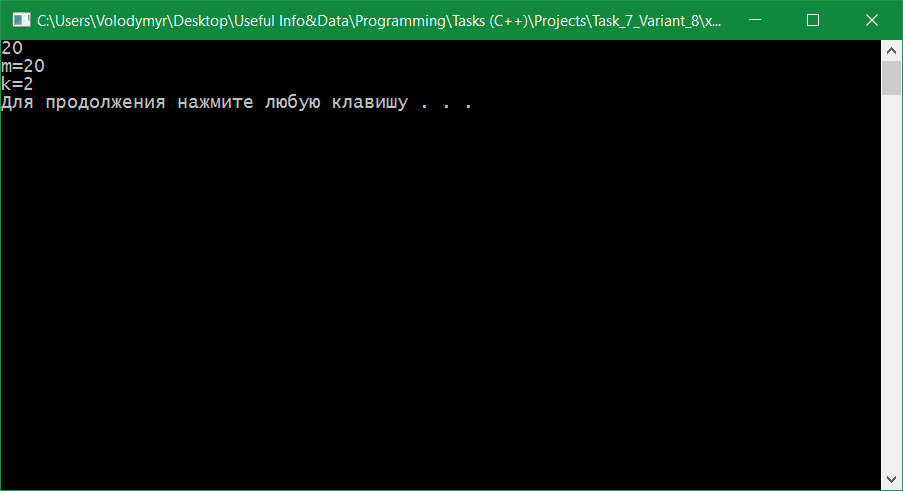
cout << "k=" << (int) k << endl;

system("pause");

return 0;

}

**Апробація**



Завдання 8

Постановка задачі

Прочитати m груп (2≤n≤10) до n в групі (4≤n≤50) цілих чисел. В кожній групі визначити, яких чисел більше: додатніх чи від’ємних та відповідно знайти суму цифр з більшої частини. Написати підпрограми (функції або/та процедури) для обчислень та перевірок вказаних властивостей, читання і друкування чисел. Декларацію підпрограм оформити у вигляді \*.h файла, а код підпрограм розмістити в окремому \*.cpp файлі. Обидва файли додати до проекту. Код основної програми повинен містити лише виклики підпрограм.

Код програми(C++)

cpp files:

1.

#include<iostream>

#include<string>

#include"allif.h"

using namespace std;

string allif(int pos, int neg, double sposfig, double snegfig)

{

if (pos > neg)

{

cout << "Sum of all figures in bigger part (positive numbers) = " << sposfig;

}

else if (neg > pos)

{

cout << "Sum of all figures in bigger part (negative numbers) = " << snegfig;

}

else

{

cout << "Quantity of positive numbers = quantity of negative numbers ( or all are zeros ) ";

}

return " ";

}

2.

#include<iostream>

#include<string>

#include"ReadAndCalculateFigures.h"

#include"allif.h"

using namespace std;

int ReadAndCalculateFigures(int m)

{

for (int i = 1; i <= m; i++)

{

int n, p, pos, neg;

double sposfig, snegfig;

cout << "How many numbers are in this group (4-50) ? ";

cin >> n;

pos = 0;

neg = 0;

sposfig = 0;

snegfig = 0;

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

{

cout << "Print a number ";

cin >> p;

if (p > 0)

{

pos = pos + 1;

while (p != 0)

{

sposfig += p % 10;

p /= 10;

}

}

if (p < 0)

{

neg = neg + 1;

while (p != 0)

{

p = abs(p);

snegfig += p % 10;

p /= 10;

}

}

}

cout << allif(pos, neg, sposfig, snegfig) << endl;

}

return 0;

}

3.

#include <iostream>

#include "ReadGroups.h"

using namespace std;

int ReadGroups()

{

int i;

cout << "Input number of groups (2-10) ";

cin >> i;

return i;

}

4.

#include<iostream>

#include<string>

#include"ReadGroups.h"

#include"allif.h"

#include"ReadAndCalculateFigures.h"

using namespace std;

int main()

{

k = ReadGroups()

ReadAndCalculateFigures(k());

system("pause");

return 0;

}

header files:

1.

#ifndef ALLIF\_H

#define ALLIF\_H

#include<string>

using namespace std;

string allif(int pos, int neg, double sposfig, double snegfig);

#endif

2.

#ifndef READGROUPS\_H

#define READGROUPS\_H

using namespace std;

int ReadGroups();

#endif

3.

#ifndef READANDCALCULATEFIGURES\_H

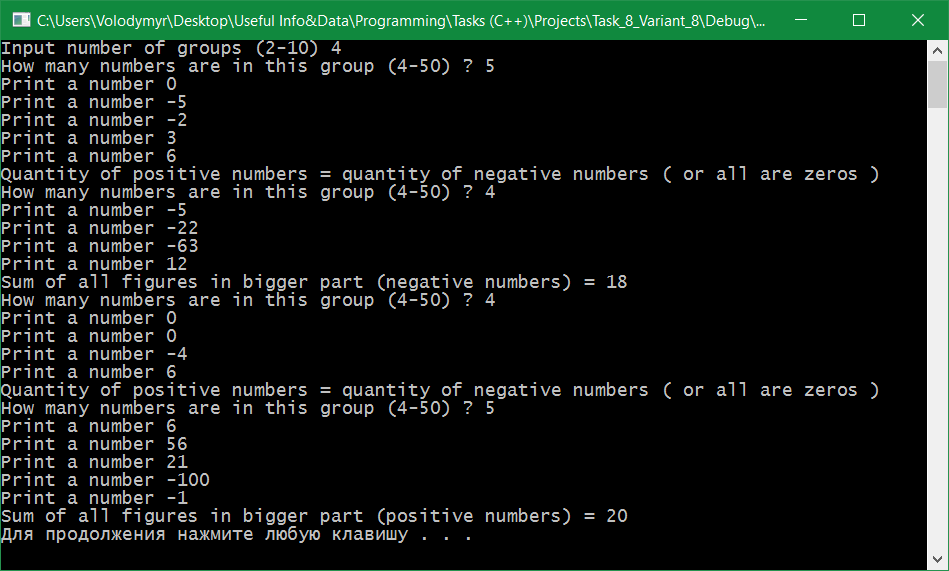
#define READANDCALCULATEFIGURES\_H

using namespace std;

int ReadAndCalculateFigures(int m);

#endif

**Апробація**



Завдання 9

Постановка задачі

Написати підпрограми (оформити з використанням \*.h файлів) для роботи зі стеком на базі двонаправленого списку. Зовні бібліотеки підпрограм повинні бути видимі лише методи запису в стек, читання зі стеку (типу PUSH та POP) та модифікації вказівника стеку (типу роботи з SP). Написати іншу бібліотеку для керування кількома стеками звертаючись при цьому до першої. Використовуючи створені підпрограми в основній програмі прочитати масив цілих чисел та вивести всі ланцюжки чисел числа, які утворюють впорядковані послідовності. При цьому послідовність ланцюжків повинна йти в прямому порядку, а числа в середині ланцюжків – в оберненому (наприклад, прочитано 1,2,3,8,3,4,5, тоді повинно бути отримано 8,3,2,1,5,4,3). Масив переглядати лише один раз.

Код програми(C++)

**Апробація**

Завдання 10

Постановка задачі

Написати підпрограми (оформити з використанням \*.h файлів) для роботи зі стеком на базі двонаправленого списку. Зовні бібліотеки підпрограм повинні бути видимі лише методи запису в стек, читання зі стеку (типу PUSH та POP) та модифікації вказівника стеку (типу роботи з SP). Написати іншу бібліотеку для керування кількома стеками звертаючись при цьому до першої. Використовуючи створені підпрограми в основній програмі прочитати масив цілих чисел та вивести всі ланцюжки чисел числа, які утворюють впорядковані послідовності. При цьому послідовність ланцюжків повинна йти в прямому порядку, а числа в середині ланцюжків – в оберненому (наприклад, прочитано 1,2,3,8,3,4,5, тоді повинно бути отримано 8,3,2,1,5,4,3). Масив переглядати лише один раз.

Код програми(C++)

**Апробація**