ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4

за курсом «Інформатика і Програмування»

студентки групи ПА-19-1

Кожан Анастасії Євгеніївни

кафедра комп’ютерних технологій, ДНУ

2019/2020 н.р.

1. **COUNTODD**
2. *Постановка задачі:*

Скласти програму CountOdd, яка:

1. Вводить з клавіатури послідовність цілих чисел.

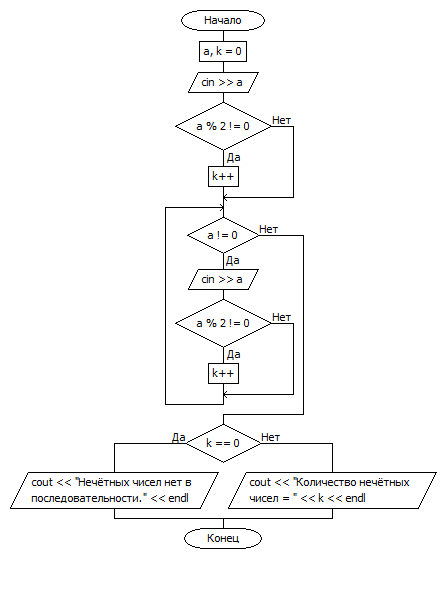
2. Рахує- скільки серед них непарних.

3. Як тільки вводиться число 0, програма припиняє роботу

    і видає результат підрахунку на екран.

1. *Опис рішення:*

Спочатку ввожу перщий елемент і за домогою if перевіряю його на непарність. Якщо непарний, то к = 0 збільшується на 1. Далі роблю цикл while, який працює до тих пір, поки користувач не введе 0. І за допомогою if перевіряю кожний елемент на непарність. Якщо k = 0, то програма видає, що елементів непарних у послідовності немає, якщо ні, то видається кількість непарних елементів.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, k=0;

cout << "Введите число = ";

cin >> a;

if (a % 2 != 0)

{

k++;

}

while (a != 0)

{

cout << "Введите число = ";

cin >> a;

if (a % 2 != 0)

{

k++;

}

}

if (k == 0)

{

cout << "Нечётных чисел нет в последовательности." << endl;

}

else

cout << "Количество нечётных чисел = " <<k<< endl;

system("pause");

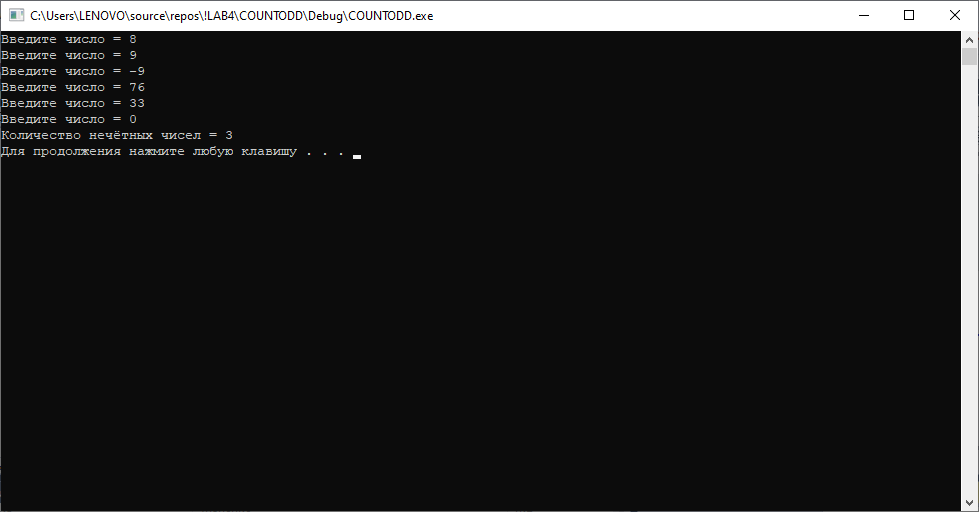
return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Програма «просить» користувача ввести послідовність чисел, яка заканічівается нулем. Користувач вводить число, а за ним - ENTER. Як тільки він введе 0, послетовательность завершітся.Программа вважає кількість непарних елементів, а потім видає його на екран. Далі користувач натискає будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **FACTOR2**
2. *Постановка задачі:*

Скласти програму FACTOR, яка для заданого натурального числа N

обчислює факторіал цього числа.

Факторіал N позначається так: N!

За визначенням N! = 1 \* 2 \* 3 \* 4 \* ... \* (N-1) \* N

Наприклад: 4! = 1 \* 2 \* 3 \* 4 = 24

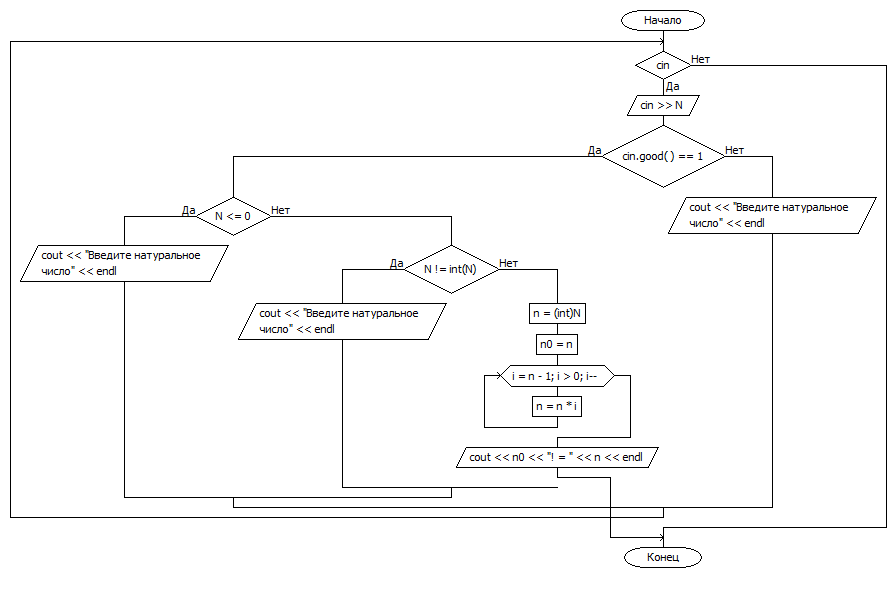
Скласти програму FACTOR2 - вдосконалений варіант програми FACTOR.

Ця програма у відповідь на неправильно введене N (тобто не натуральне число)

видає зрозумілу підказку для користувача.

1. *Опис рішення:*

За допомогою if перевіряю введене на валідність вводу. Далі за допомогою цикла for отримую факторіал введеного числа і вивожу його на екран.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, n0 ;

float N;

cout << "Введите число = ";

while (cin) {

cin >> N;

if (cin.good() == 1)

{

if (N <= 0)

{

cout << "Введите натуральное число" << endl;

}

else if (N != int(N))

cout << "Введите натуральное число" << endl;

else

{

n = (int)N;

n0 = n;

for (int i = n - 1; i > 0; i--)

{

n = n \* i;

}

cout << n0 << "! = " << n << endl;

break;

}

}

else

cout << "Введите натуральное число" << endl;

}

system("pause");

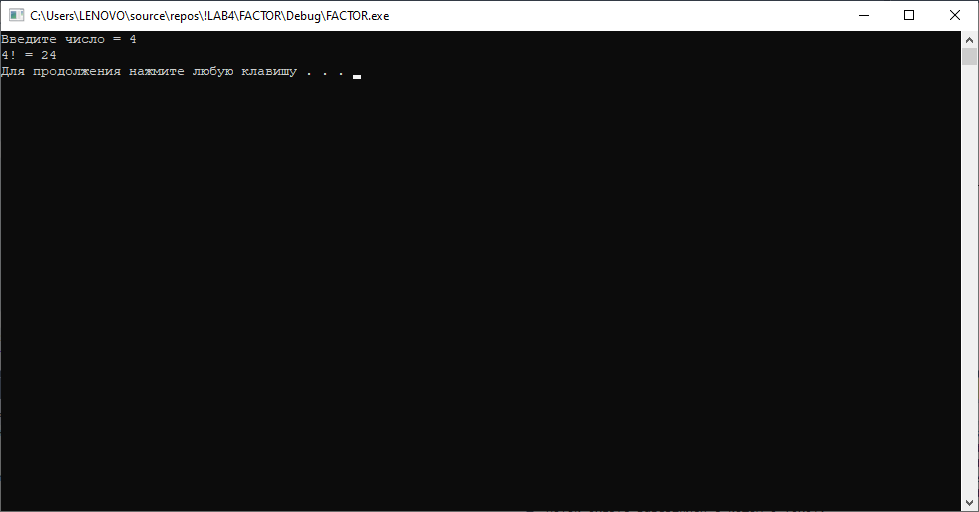
return 0;

}

1. *Описан інтерфейсу:*

Користувач запускає програму. Програма «просить» користувача ввести натурально число. Потім програма перевіряє на правильність введення. Якщо введення неправильний, то видається відповідне повідомлення. Якщо немає, то програма вважає факторіал введеного числа. Далі користувач натискає будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



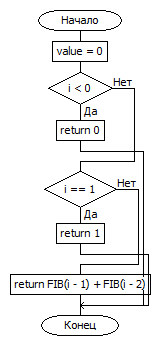
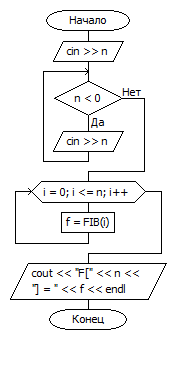
1. **FIBONACCI**
2. *Постановка задачі:*

Складіть програму FIBO, яка:

вводить з клавіатури число N, обчислює число Фібоначчі з номером N (тобто f [N])

і виводить його на екран.

1. *Опис рішення:*



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int FIB(int i)

{

int value = 0;

if (i < 0)

return 0;

if (i == 1)

return 1;

return FIB(i - 1) + FIB(i - 2);

}

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, f;

cout << "Введите индекс элемента n = ";

cin >> n;

while (n < 0)

{

cout << "Введите неотрицательное целое число = ";

cin >> n;

}

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

{

f = FIB(i);

}

cout << "F[" << n << "] = " << f << endl;

system("pause");

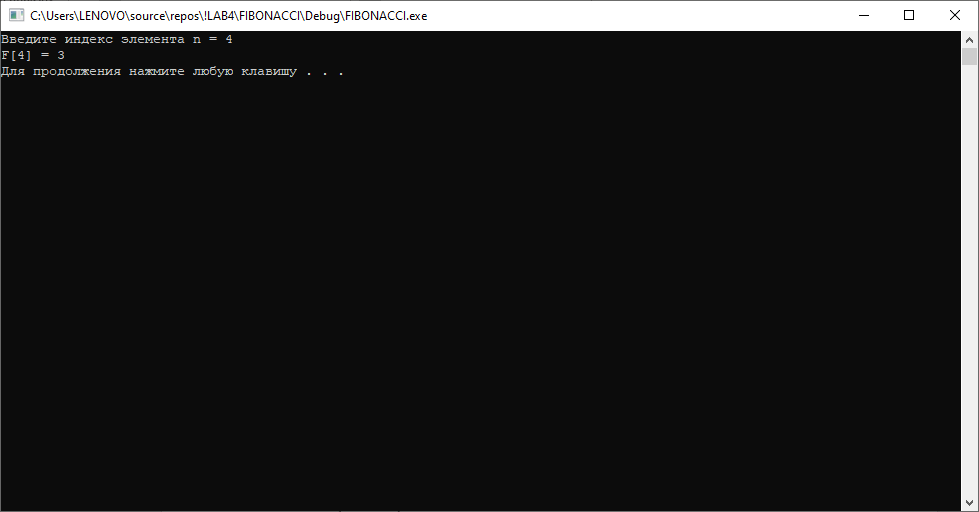
return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Програма «просить» користувача ввести невід'ємне число. Як тільки користувач ввів неотрицательно число, програма видає число Фібоначчі. Далі користувач натискає будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **HIT2**
2. *Постановка задачі:*

Скласти прогамму HIT2, яка:

0. "Задумує" окружність O з центром в точці (x0, y0)

   і радіусом R.

1. Вводить з клавіатури речові координати (x, y)

   точки A на площині.

2. Перевіряє, чи потрапляє точка A всередину (або на кордон)

   кола O.

3. До тих пір, поки точка не потрапила всередину,

   виводить повідомлення

   "Найтепліше" або "Найхолодніше" в залежності від того, ближче або

   далі від центру кола потрапила нова точка в порівнянні

   з попередньої,

   і повторює пункти 1, 2, 3.

4. Виводить результат на екран у вигляді:

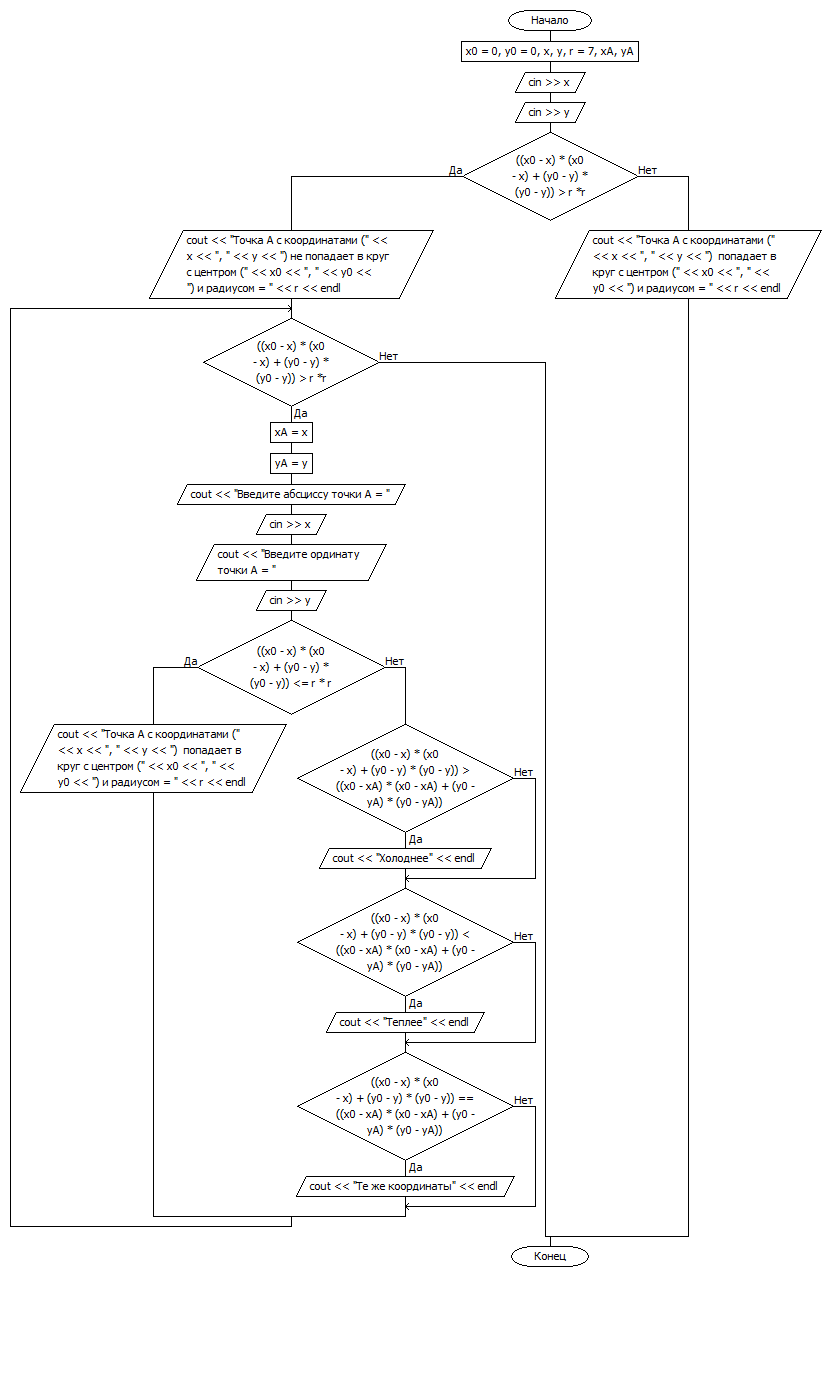
   "Точка (x, y) потрапляла в коло з центром в точці (x0, y0)

    радіуса R ",

    підставляючи в результат замість R, x0, y0, x, y їх значення.

1. *Опис рішення:*

Спочатку я задаю константові змінні центра кола та радіуса, далі користувач повинен ввести координати точка А. за допомогою цикла while,який працює, поки точка не попаде у коло, я перевіряю, чи входить точка в коло за допомогою формули. За допомогою if перевіряю, якщо точка знаходиться ближче, ніж минулого разу, то програма видає «Тепліше», якщо далі - то «Холодніше», якщо координати ті самі, то програма видає: «Ті самі координати». Коли користувач введе координати, які будуть входити у коло, то програма видасть, що точка входить у коло і на цьому завершить роботу.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int x0 = 0, y0 = 0, x, y, r= 7, xA, yA;

cout << "Введите абсциссу точки А = ";

cin >> x;

cout << "Введите ординату точки А = ";

cin >> y;

if (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) > r \* r)

{

cout << "Точка А с координатами (" << x << ", " << y << ") не попадает в круг с центром (" << x0 << ", " << y0 << ") и радиусом = " << r << endl;

while (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) > r \* r)

{

xA = x;

yA = y;

cout << "Введите абсциссу точки А = ";

cin >> x;

cout << "Введите ординату точки А = ";

cin >> y;

if (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) <= r \* r)

cout << "Точка А с координатами (" << x << ", " << y << ") попадает в круг с центром (" << x0 << ", " << y0 << ") и радиусом = " << r << endl;

else

{

if (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) > ((x0 - xA)\*(x0 - xA) + (y0 - yA)\*(y0 - yA)))

cout << "Холоднее" << endl;

if (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) < ((x0 - xA)\*(x0 - xA) + (y0 - yA)\*(y0 - yA)))

cout << "Теплее" << endl;

if (((x0 - x)\*(x0 - x) + (y0 - y)\*(y0 - y)) == ((x0 - xA)\*(x0 - xA) + (y0 - yA)\*(y0 - yA)))

cout << "Те же координаты" << endl;

}

}

}

else

cout << "Точка А с координатами (" << x << ", " << y << ") попадает в круг с центром (" << x0 << ", " << y0 << ") и радиусом = " << r << endl;

system("pause");

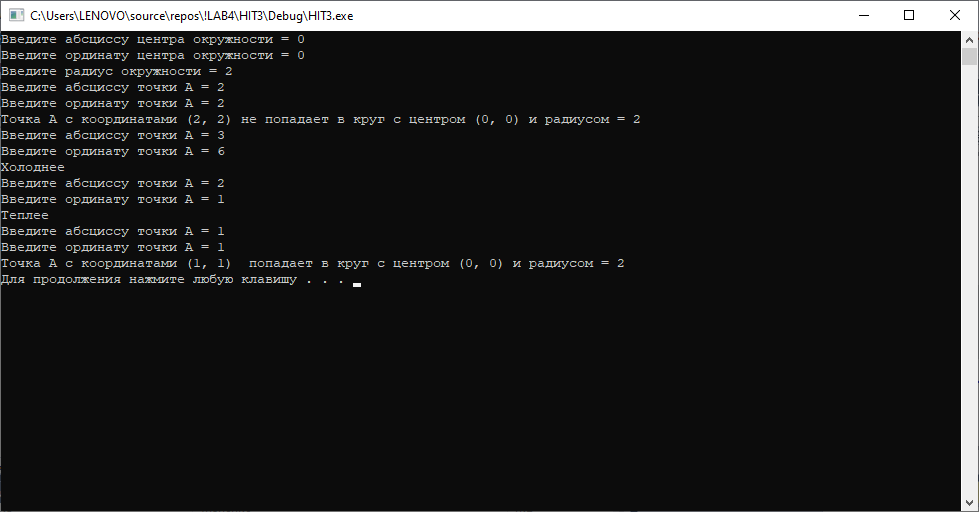
return 0;

}

1. *Описа інтерфейсу:*

Користувач запускає програму. Програма «просить» ввести певні координати центру кола, точки А і радіус. Як тільки користувач все дані введе і натисне ENTER, програма вважає, входить точка А в коло або його межі. Якщо немає, форма видає «тепліше» або «холодніше», в залежності від попереднього результату, поки користувач не потрапить в область кола. Далі користувач натискає будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **MAX**
2. *Постановка задачі:*

Скласти програму MAX, яка:

1) вводить з клавіатури послідовність цілих чисел,

    що закінчується нулем;

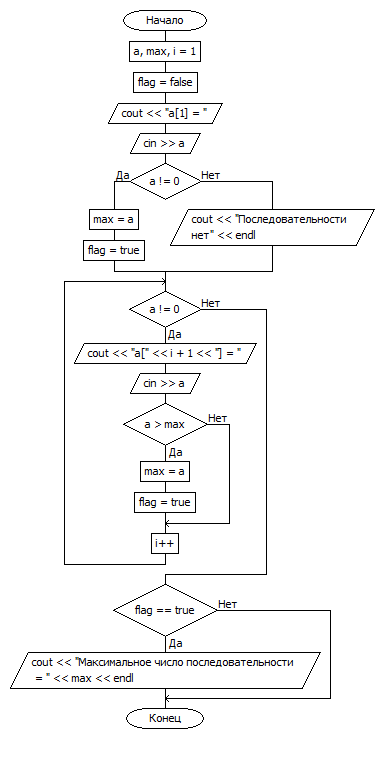
2) знаходить найбільший з усіх членів послідовності;

3) виводить на екран знайдене значення в вигляді:

    "MAX = ...".

1. *Опис рішення:*

Спочатку ввожу перщий елемент і за домогою максимальному значенню присваюю переше значення послідовності. Далі роблю цикл while, який працює до тих пір, поки користувач не введе 0 і у цей час І за допомогою if перевіряю кожний елемент, чи більше він за поточне максимальне значення Якщо так, то присваюю максимальному значенню значення поточного елемента. Далі програма виводить максимальне значення на екран.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, max, i = 1;

bool flag = false;

cout << "Введите последовательность целых чисел. Ноль - признак конца." << endl;

cout << "а[1] = ";

cin >> a;

if (a != 0)

{

max = a;

flag = true;

}

else

{

cout << "Последовательности нет" << endl;

}

while (a != 0)

{

cout << "a[" << i + 1 << "] = ";

cin >> a;

if (a > max)

{

max = a;

flag = true;

}

i++;

}

if (flag == true)

{

cout << "Максимальное число последовательности = " << max << endl;

}

system("pause");

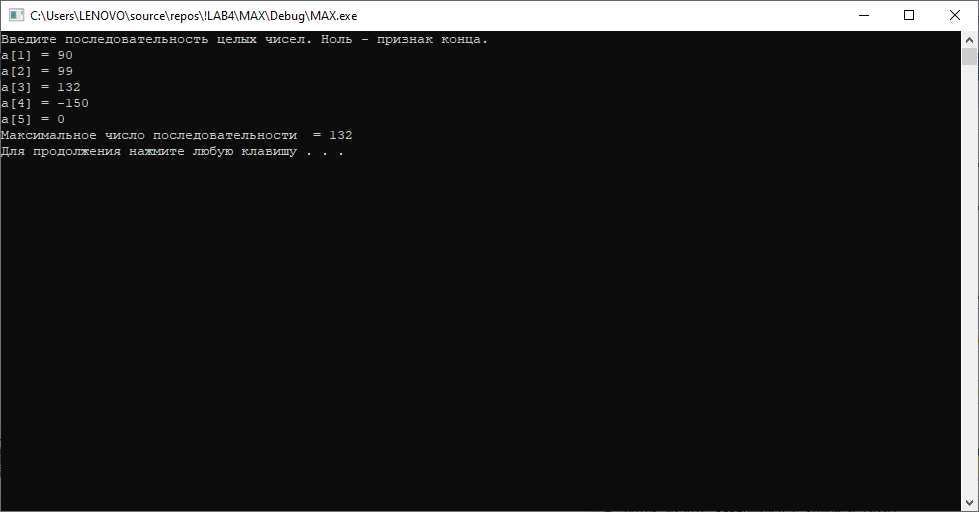
return 0;

}

1. *Опис интерфейсу:*

Користувач запускає програму. Форма запитує ввести послідовність, яка закінчується нулем. Користувач вводить число, а за ним - ENTER. Як тільки він введе 0, послідовність завершиться. Далі програма знаходить максимально число, і коли користувач завершить послідовність і натисне ENTER, програма видасть максимальне число послідовності. Потім користувачеві потрібно натиснути будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **MENU2**
2. *Постановка задачі:*

Скласти програму MENU2, яка:

1) виводить на екран наступний текст:

    "Працює інформаційна система СШ 23

     1. Новини дня

     2. Анекдот тижня

     3. Автора !!!

     0. Вихід

     Введіть відповідний номер і натисніть ENTER. "

2) вводить з клавіатури номер;

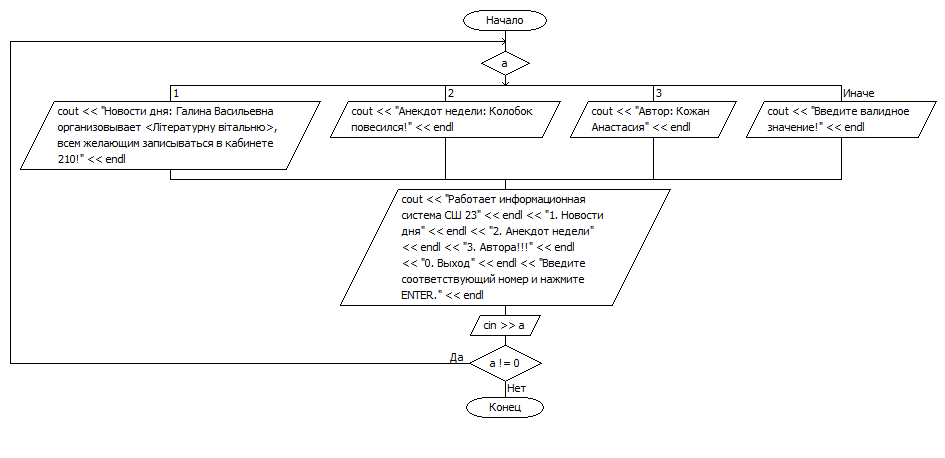
3) виводить на екран відповідну інформацію;

4) продовжує циклічно виконувати вищевказані дії

    до тих пір, поки користувач не вибере "Вихід".

1. *Опис рішения:*

Спочатку користувач має ввести число (1, 2 чи 3). За допомогою цикла do while (працює, поки користувач не введе 0) перевіряю значення, що ввів користувач за допомогою switch і case. Видаю відповідні результати, якщо користувач ввів невалідне значення, то програма видає мессендж ввести валідне. Якщо користувач вводить 0, програма виходить з циклу і завершує роботу.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a;

cout << "Работает информационная система СШ 23" << endl << "1. Новости дня" << endl << "2. Анекдот недели" << endl << "3. Автора!!!" << endl << "0. Выход" << endl << "Введите соответствующий номер и нажмите ENTER." << endl;

cin >> a;

do {

switch (a)

{

case 1:

cout << "Новости дня: Галина Васильевна организовывает <Літературну вітальню>, всем желающим записываться в кабинете 210!" << endl;

break;

case 2:

cout << "Анекдот недели: Колобок повесился!" << endl;

break;

case 3:

cout << "Автор: Кожан Анастасия" << endl;

break;

default: cout << "Введите валидное значение!" << endl;

}

cout << "Работает информационная система СШ 23" << endl << "1. Новости дня" << endl << "2. Анекдот недели" << endl << "3. Автора!!!" << endl << "0. Выход" << endl << "Введите соответствующий номер и нажмите ENTER." << endl;

cin >> a;

}

while (a != 0);

system("pause");

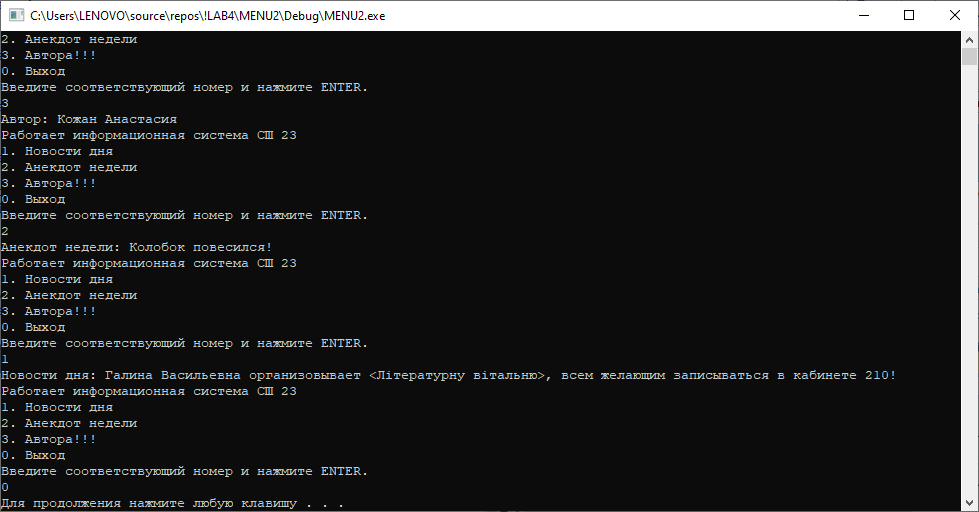
return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Форма надає вибір між трьома пунктами. Користувач вибирає і вводить відповідний номер. Далі програма видає відповідне повідомлення. Дії повторюються, поки користувач не введе 0 - вихід з циклічності. Потім користувачеві потрібно натиснути будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **Parrot2**
2. *Постановка задачі:*

Складіть програму PARROT2, яка:

1. Просить користувача ввести з клавіатури

    один рядок якогось тексту.

2. Виводить цей рядок на екран.

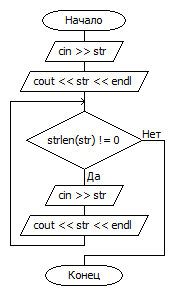
3. Повторює ці дії до тих пір,

    поки користувач не введе порожній рядок

    (Тобто не натисне просто ENTER).

1. *Опис рішення:*

Користувач вводить якусь строку. Програма видає її обратно. За допомогою цикла while, який працює, поки користувач не введе порожній рядок, повторюю дії. Програма вийде з цикла, коли strlen(str)!=0 і завершить свою роботу.



1. *Код програми:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char str[35];

cout << "Введите строку из какого-то текста: " << endl;

cin.getline(str, 34);

cout << str << endl;

while (strlen(str)!=0)

{

cout << "Введите строку из какого-то текста: " << endl;

cin.getline(str, 34);

cout << str << endl;

}

system("pause");

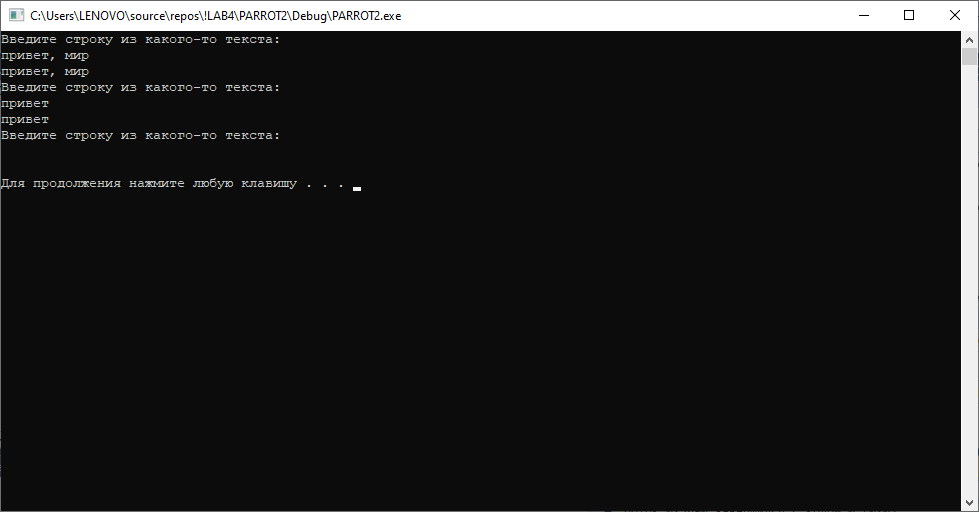
return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Форма запитує ввести рядок з тексту. Потім програма «повторює» рядок. Програма буде запитувати ввести рядок і повторювати еёдо тих пір поки, користувач не введе ENTER. Потім користувачеві потрібно натиснути будь-яку клавішу і програма завершить свою роботу.

1. *Опис тестових прикладів:*



1. **PRODUCT**
2. *Постановка задачі:*

Скласти програму PRODUCT, яка:

1) вводить з клавіатури послідовність цілих чисел,

    що закінчується нулем;

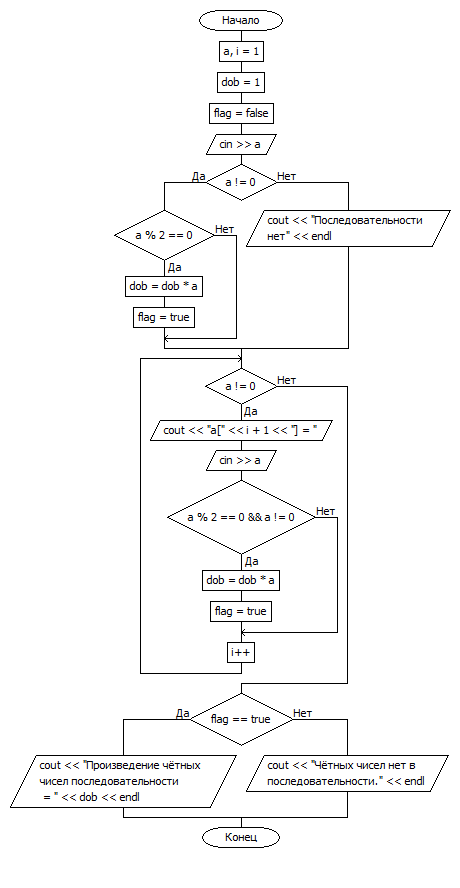
2) знаходить добуток всіх парних членів послідовності;

3) виводить на екран обчислений добуток у вигляді:

    "ДОБУТОК = ...".

1. *Опис рішення:*

Спочатку ввожу перший елемент і за допомогою if перевіряю його на парність. Якщо парний, то double dob = 1 буде dob = dob\*a і булева змінна bool flag = false становиться flag = true. Далі роблю цикл while, який працює до тих пір, поки користувач не введе 0. І за допомогою if перевіряю кожний елемент на парність. Якщо парний, то dob = dob\*a. Коли користувач вводить нуль, програма виходить з циклу і за допомогою if перевіряю, чи правдива змінна flag. Якщо так, то вивожу добуток парних чисел послідовності. Якщо ні, то програма видає, що парних чисел у послідовності немає.



1. *Код програми:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, i = 1;

double dob = 1;

bool flag = false;

cout << "Введите последовательность целых чисел. Ноль - признак конца." << endl;

cout << "а[1] = ";

cin >> a;

if ( a!=0)

{

if (a % 2 == 0)

{

dob = dob \* a;

flag = true;

}

}

else

{

cout << "Последовательности нет" << endl;

}

while (a != 0)

{

cout << "a[" << i + 1 << "] = ";

cin >> a;

if (a % 2 == 0 && a != 0 )

{

dob = dob \* a;

flag = true;

}

i++;

}

if (flag == true)

{

cout << "Произведение чётных чисел последовательности = " << dob << endl;

}

else

cout << "Чётных чисел нет в последовательности." << endl;

system("pause");

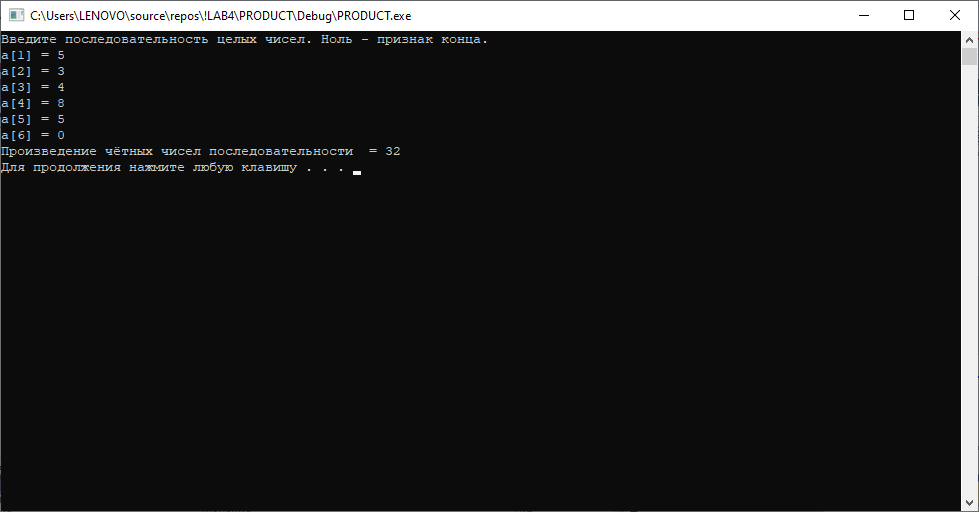
return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Форма запитує ввести послідовність, яка закінчується нулем. Користувач вводить число, а за ним - ENTER. Як тільки він введе 0, послідовність завершиться. Далі програма знаходить твір парних чисел послідовності і виводить його на екран, якщо парних числі немає, то програма виводить на екран фразу: «парних чисел в послідовності немає». Потім користувачеві потрібно натиснути будь-яку клавішу і програма завершить свою роботу

1. *Опис тестових прикладів:*



1. **SUM**
2. *Постановка задачі:*

*Скласти програму SUM, яка:*

*1) вводить з клавіатури послідовність цілих чисел,*

*що закінчується нулем;*

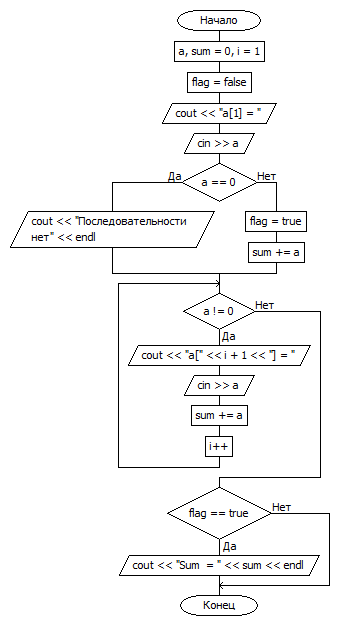
*2) знаходить суму всіх членів послідовності;*

*3) виводить на екран знайдене значення в вигляді:*

*"SUM = ...".*

1. *Опис рішення:*

Спочатку задаю цілі змінні (sum = 0, a)Роблю цикл while, який працює до тих пір, поки користувач не введе 0. Якщо користувач вводить першим нуль, то за допомогою if програма видає, що послідовності немає. Якщо ні, то програма рахую суму усіх елементів (sum +=a). Після того, як програма вийде з циклу, на екран виводиться сума усіх елементів та завершується робота програми.



1. *Код программи:*

#include "pch.h"

#include <iostream>

#include <Windows.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a, sum=0, i = 1;

bool flag = false;

cout << "Введите последовательность целых чисел. Ноль - признак конца." << endl;

cout << "а[1] = ";

cin >> a;

if (a == 0)

{

cout << "Последовательности нет" << endl;

}

else

{

flag = true;

sum += a;

}

while (a != 0)

{

cout << "a[" << i + 1 << "] = ";

cin >> a;

sum += a;

i++;

}

if (flag == true)

{

cout << "Sum = " << sum << endl;

}

system("pause");

return 0;

}

1. *Опис інтерфейсу:*

Користувач запускає програму. Форма запитує ввести послідовність, яка закінчується нулем. Користувач вводить число, а за ним - ENTER. Як тільки він введе 0, послідовність завершиться. Далі програма знаходить суму всіх його елементів і виводить її на екран, якщо ж елементів немає, то на екран виводиться фраза: «Послідовності немає». Потім користувачеві потрібно натиснути будь-яку клавішу і програма завершить свою роботу

1. *Опис тестових прикладів:*