МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЗВІТ**

**з лабораторної роботи № 1**

**«Алгоритми з розгалуженням та комбіновані алгоритми**

**121 – Інженерія програмного забезпечення. 2021–2022 н.р.»**

# Дисципліна «Алгоритми та структури даних 1. Основи алгоритмізації»

**Виконала :**

студентка 1-го курсу, групи КП-11,

спеціальності 121 – Інженерія

програмного забезпечення

*Кирильчук Олександра Артурівна*

Київ – 2021

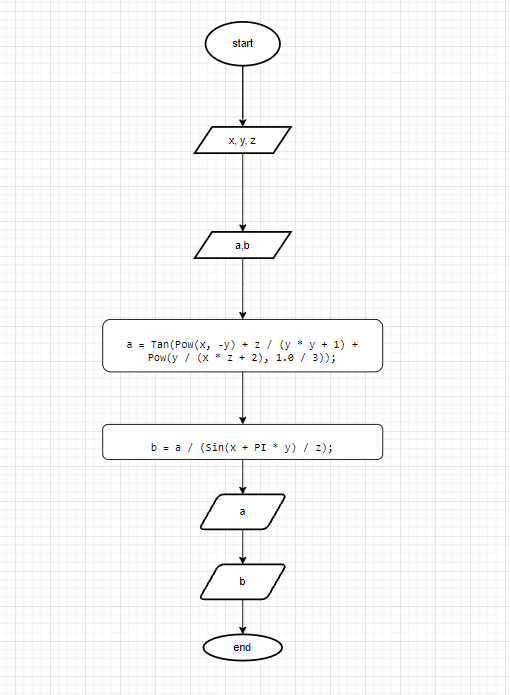
**Варіант**



**Задача 1**

****





Код:

using System;

using static System.Math;

using static System.Console;

class MathExample

{

static void Main(string[] args)

{

double x, y, z, a, b;

Write("x = "); x = Convert.ToDouble(ReadLine());

Write("y = "); y = Convert.ToDouble(ReadLine());

Write("z = "); z = Convert.ToDouble(ReadLine());

if ((y \* y + 1) != 0 && (x \* z + 2) != 0)

{

a = Tan(Pow(x, -y) + z / (y \* y + 1) + Pow(y / (x \* z + 2), 1.0 / 3));

WriteLine("a = {0}", a);

if ((Sin(x + PI \* y) / z) != 0)

{

b = a / (Sin(x + PI \* y) / z);

WriteLine("b = {0}", b);

}

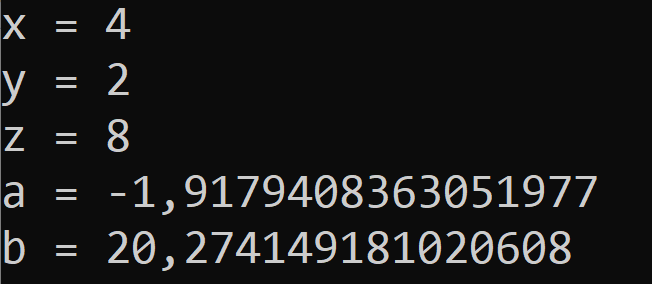
}

ReadKey();

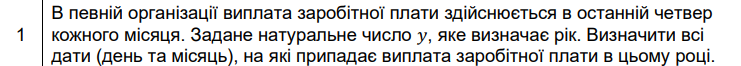
}

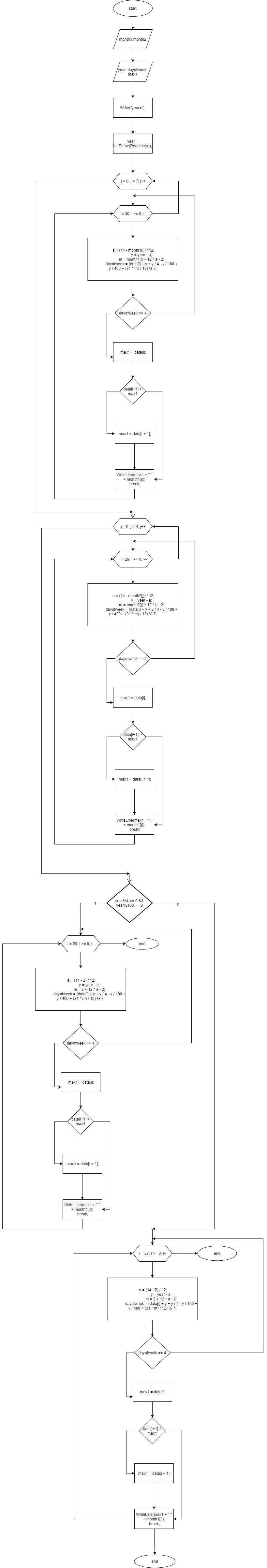
}

Результат:



**Задача 2**





Код:

using System;

using static System.Math;

using static System.Console;

class MathExample

{

static int[] month1 = new int[7]

{1, 3, 5, 7, 8, 10, 12 };

static int[] month2 = new int[4]

{4, 6, 9, 11};

static int[] data = new int[31]

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 };

static void Main(string[] args)

{

int year, dayofweek, max1;

Write("year="); year = int.Parse(ReadLine());

for (int j = 0; j < 7; j++)

{

for (int i = 30; i >= 0; i--)

{

int a = (14 - month1[j]) / 12;

int y = year - a;

int m = month1[j] + 12 \* a - 2;

dayofweek = (data[i] + y + y / 4 - y / 100 + y / 400 + (31 \* m) / 12) % 7;

if (dayofweek == 4)

{

max1 = data[i];

if(data[i+1] > max1)

{

max1 = data[i + 1];

}

WriteLine(max1 + "." + month1[j]);

break;

}

}

}

for (int j = 0; j < 4; j++)

{

for (int i = 29; i >= 0; i--)

{

int a = (14 - month2[j]) / 12;

int y = year - a;

int m = month2[j] + 12 \* a - 2;

dayofweek = (data[i] + y + y / 4 - y / 100 + y / 400 + (31 \* m) / 12) % 7;

if (dayofweek == 4)

{

max1 = data[i];

if (data[i + 1] > max1)

{

max1 = data[i + 1];

}

WriteLine(max1 + "." + month2[j]);

break;

}

}

}

if (year%4 == 0 && year%100 == 0)

{

for (int i = 28; i >= 0; i--)

{

int a = (14 - 2) / 12;

int y = year - a;

int m = 2 + 12 \* a - 2;

dayofweek = (data[i] + y + y / 4 - y / 100 + y / 400 + (31 \* m) / 12) % 7;

if (dayofweek == 4)

{

max1 = data[i];

if (data[i + 1] > max1)

{

max1 = data[i + 1];

}

WriteLine(max1 + ".2");

break;

}

}

}

else

{

for (int i = 27; i >= 0; i--)

{

int a = (14 - 2) / 12;

int y = year - a;

int m = 2 + 12 \* a - 2;

dayofweek = (data[i] + y + y / 4 - y / 100 + y / 400 + (31 \* m) / 12) % 7;

if (dayofweek == 4)

{

max1 = data[i];

if (data[i + 1] > max1)

{

max1 = data[i + 1];

}

WriteLine(max1 + ".2");

break;

}

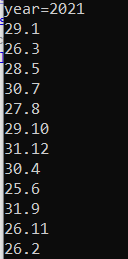
}

}

}

}

Результат:



**Контрольні запитання**

1. **Що таке алгоритм?**

Алгоритм – це скінченна послідовність однозначних дій, яку необхідно виконати для розв’язання поставленої задачі та досягнення очікуваного результату.

1. **Які є властивості алгоритму?**

* Точність ( Алгоритм має містити однозначно визначену послідовність однозначно сформульованих дій (кроків))
* Скінченність ( Алгоритм має завершитись після виконання визначеної кількості кроків за скінченний проміжок часу)
* Здійсненність (Алгоритм повинен бути здійсненим в умовах реального світу. Він не може бути абстрактним або уявним
* Результативність (Алгоритм завжди повинен завершуватись отриманням кінцевого результату, який має відповідати поставленій задачі)
* Масовість (Алгоритм має бути застосованим до всього класу задач такого типу, які відрізняються лише початковими даними)
* Ефективність (Алгоритм повинен забезпечувати розв’язання задачі за мінімальний час із мінімальними ресурсами).

1. **Які існують керуючі конструкції?**

Три фундаментальні керуючі конструкції

* Конструкція слідування (Sequence) - передбачає порядко́ве виконання кроків алгоритму у тому порядку, в якому вони з’являються в алгоритмі.
* Конструкція розгалуження (Selection, Decision) - використовується для визначення, який крок або групу кроків необхідно виконати шляхом вибору одного шляху, що залежить від умови.
* Конструкція повторення (Repetition, Iteration, Loop) - дозволяє певній частині алгоритму бути виконаною довільну кількість разів, залежно від поставленої умови. Інша назва конструкції повторення – це цикл

( Існує 3 типи циклів: ▪ цикл з передумовою ▪ цикл з післяумовою ▪ цикл з лічильником).

* Рекурсія (Recursion).

1. **Які існують типи алгоритмів?**

* Послідовний (лінійний) алгоритм
* Розгалужений алгоритм
* Циклічний алгоритм
* Комбінований алгоритм

1. **З яких етапів складається розв’язання задачі на комп’ютері?**

* Постановка задачі
* Побудова математичної моделі (за потреби)
* Алгоритмізація поставленої задачі
* Програмування за складеним алгоритмом
* Тестування та налагодження програми