

# Розвиваємо алгоритмічне мислення

## Алгоритми з розгалуженням

*Нижче наведені умови задач та їх розв'язки, оформлені у вигляді блок-схем. Для першої задачі наведено програмну реалізацію запропонованого алгоритму.*

**Вам необхідно проаналізувати ці задачі та алгоритми і скласти програми їх вирішення, використовуючи наведені блок-схеми.**

*Пропонуємо самостійно протестувати правильність складених програм за допомогою наведених прикладів вхідних даних та результатів виконання програм для цих даних.*

*Це завдання не оцінюється і не впливає на підсумкову оцінку за курс та отримання сертифікату.*

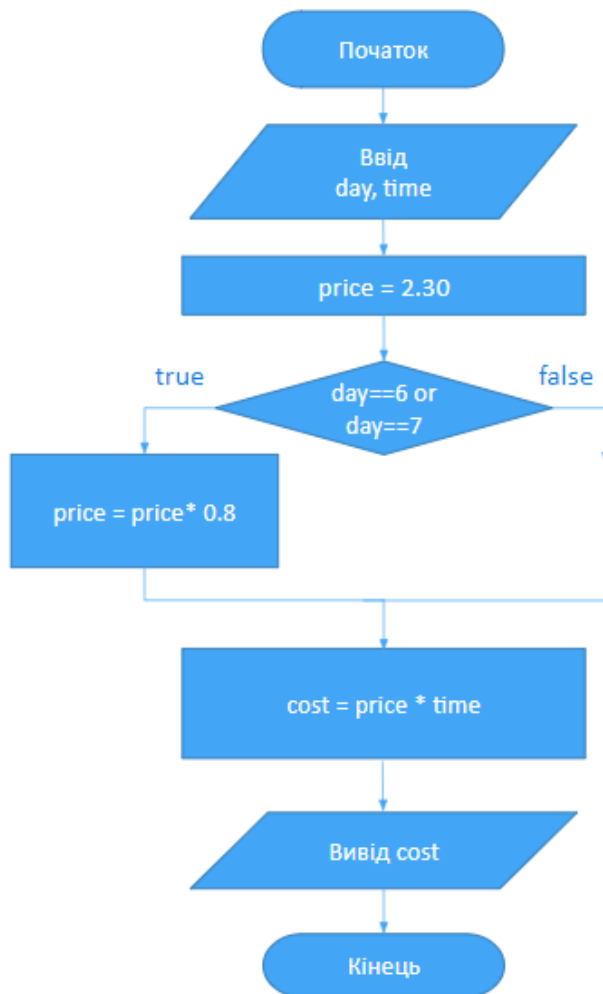
### Задача 1.

Створіть програму визначення вартості дзвінка по телефону. Ціна за хвилину розмови 2.30 грн. У суботу та у неділю надається знижка 20%.

#### Пояснення розв'язку

*Ввести номер дня тижня `day` та тривалість розмови `time`. Задано ціну за хвилину розмови `price = 2.30`. Порівнюємо номер дня тижня з числом 6 (субота) та числом 7 (неділя), якщо день тижня виявиться рівним 6 або 7, то обчислюємо ціну зі знижкою  $price = price * 0.8$  (без знижки коефіцієнт був би рівний 1). Якщо ж умова не виконується то `price` залишається без змін. Далі шукаємо добуток `price` та `time` і заносимо в змінну `cost`. Виводимо `cost` на екран.*

**Блок-схема:**



Код:

```

day = int(input())
time = int(input())
price = 2.30
if (day==6 or day ==7):
    price = price * 0.8
cost= price*time
print(cost)

```

Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
1 10	23.0
6 20	36.8

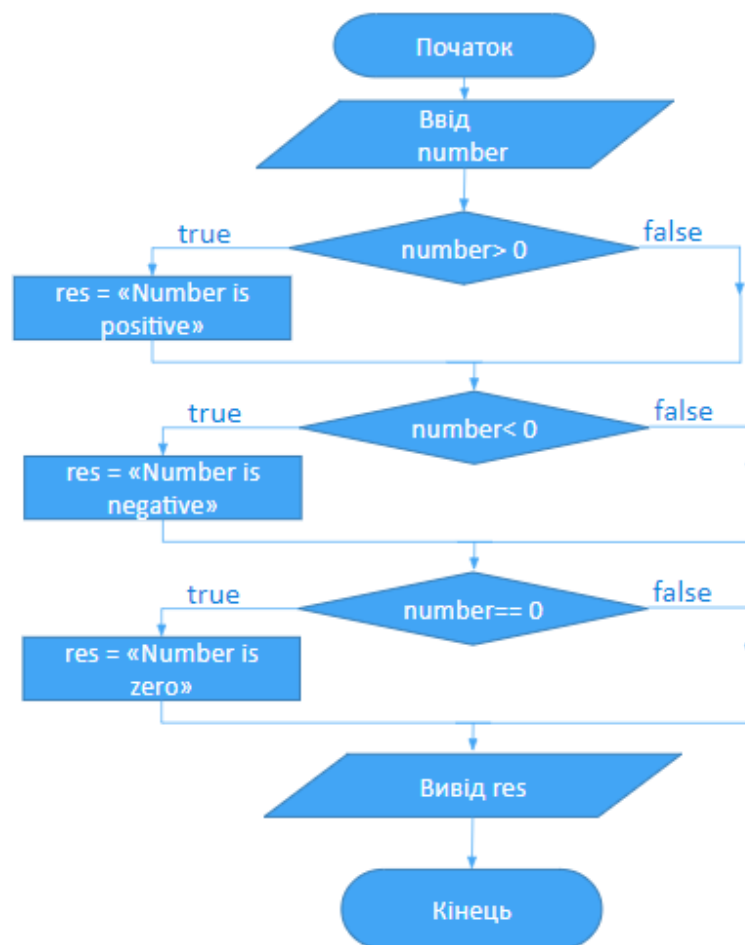
## Задача 2.

Напишіть програму яка перевіряє, чи є введене число додатним, від'ємним або це нуль.

### Пояснення розв'язку

Вводимо число *number*. Порівнюємо його з нулем. Якщо  $number > 0$ , то *res* присвоюємо "Number is positive". Далі перевіряємо чи  $number < 0$ , якщо так, то *res* присвоюємо "Number is negative". Далі перевіряємо чи  $number == 0$ , якщо так, то *res* присвоюємо "Number is zero". Виводимо значення змінної *res*.

### Блок-схема:



### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10	Number is positive
0	Number is zero
-4	Number is negative

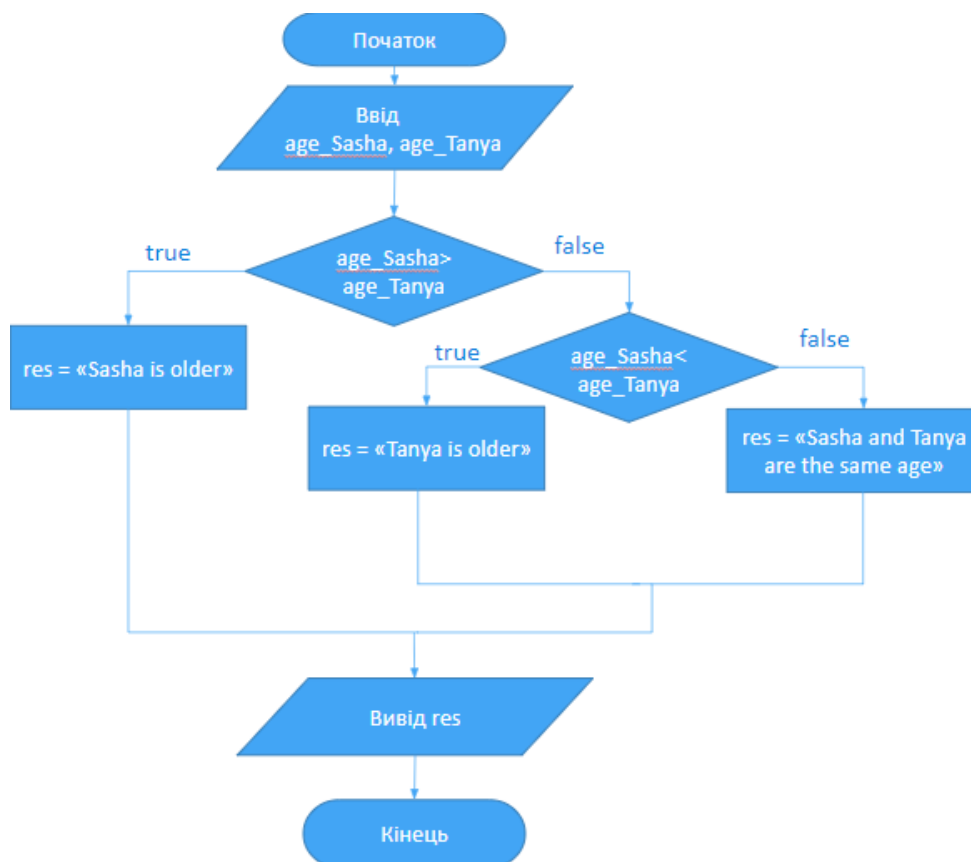
### Задача 3.

Напишіть програму, на вхід якої подається два цілих числа - вік Сашка і вік Тетянки. Програма має вивести повідомлення про те, хто є старшим серед них.

#### Пояснення розв'язку

Вводимо два цілих числа  $age\_Sasha$  та  $age\_Tanya$ . Порівнюємо їх. Якщо виявиться, що  $age\_Sasha > age\_Tanya$ , то змінній  $res$  присвоюємо **"Sasha is older"**. Далі перевіряємо чи  $age\_Sasha < age\_Tanya$ . Якщо ж  $age\_Sasha < age\_Tanya$ ,  $res$  присвоюємо **"Tanya is older"**. У випадку, якщо жодна вимога не виконується, це означає, що вік обидвох однаковий, а отже  $res$  присвоюємо **"Sasha and Tanya are the same age"**.

#### Блок-схема:



#### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10 15	<b>Tanya is older</b>
50 25	<b>Sasha is older</b>
50 50	<b>Sasha and Tanya are the same age</b>

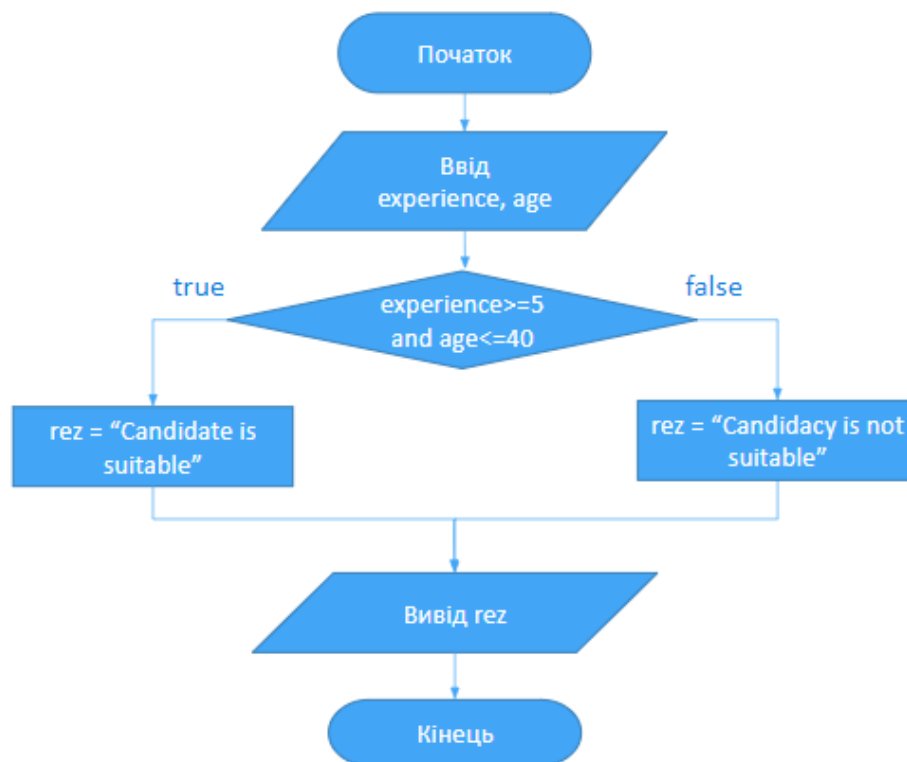
## Задача 4.

Підприємство набирає робітників у штат. Умова прийому потребує не менше 5 років робочого стажу та вік не більше 40 років. Створіть програму, яка перевіряє, підходить кандидатура по цим параметрам чи ні.

### Пояснення розв'язку

Вводимо стаж робітника *experience* та його вік *age*. Виконуємо порівняння  $experience \geq 5$  і одночасно чи  $age \leq 40$  (для цього використовуємо логічну операцію "і"), якщо обидві умови виконуються одночасно то змінній *rez* присвоюємо "Candidate is suitable", інакше присвоюємо їй "Candidacy is not suitable"

### Блок-схема:



### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10 20	Candidate is suitable
1 25	Candidacy is not suitable
10 50	Candidacy is not suitable

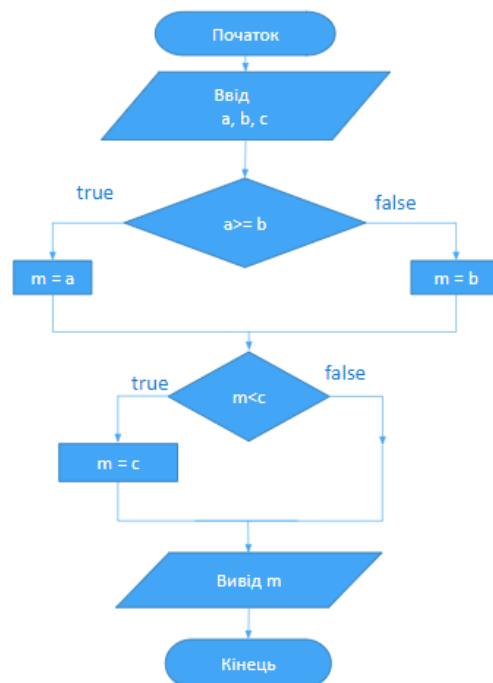
## Задача 5.

Дано три числа  $a$ ,  $b$ ,  $c$ . Змінній  $m$  присвоїти значення найбільшого з трьох чисел.

### Пояснення розв'язку

Вводимо три числа  $a$ ,  $b$ ,  $c$ . Спочатку знаходимо більше з двох чисел: порівнюємо  $a$  та  $b$ , якщо  $a \geq b$ , то  $m$  присвоюємо значення змінної  $a$ , інакше  $m$  присвоюємо значення  $b$ . Далі порівнюємо  $m$  з третім числом  $c$ , якщо  $m < c$ , то  $m$  присвоюємо значення  $c$ , інакше не виконуємо додаткових дій. В результаті виводимо значення змінної  $m$ .

### Блок-схема:



### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10 20 30	30
10 15 5	15
10 5 20	20

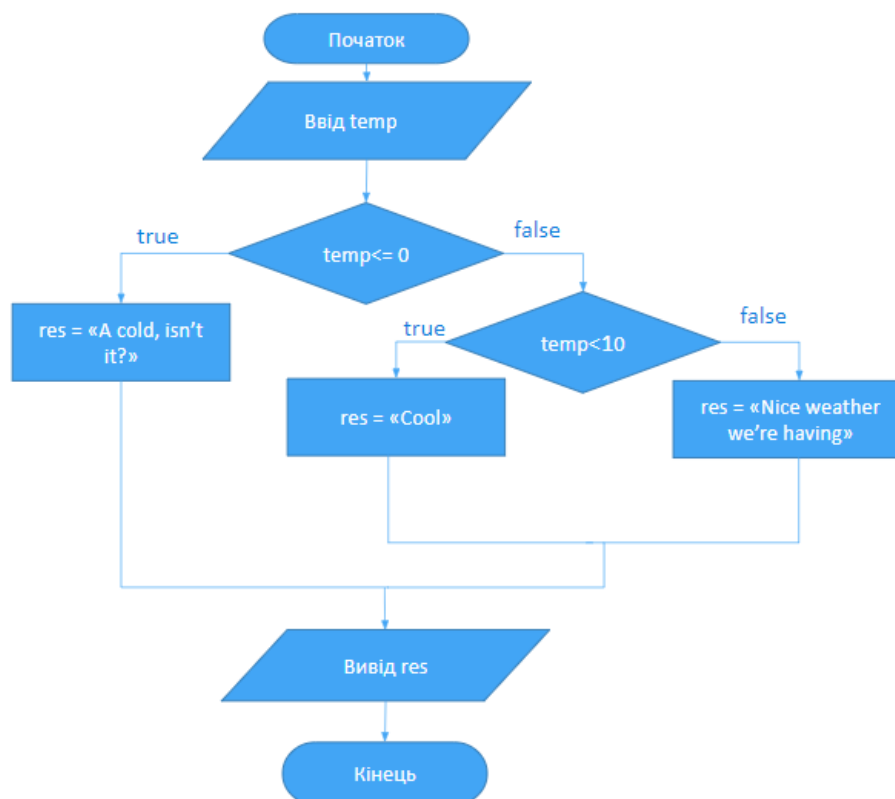
## Задача 6.

Напишіть програму, в якій користувач вводить значення температури в градусах Цельсія, і, якщо це значення менше або дорівнює 0, необхідно вивести повідомлення "A cold, isn't it?". Якщо ж температура становить більше 0 і менше 10 градусів Цельсія повідомлення буде "Cool", у інших випадках "Nice weather we're having".

### Пояснення розв'язку

Вводимо значення змінної *temp*. Порівнюємо це значення з нулем, якщо  $temp \leq 0$  то змінній *res* присвоюємо значення "A cold, isn't it?". Далі, якщо умова не виконується перевіряємо чи  $temp < 10$ , якщо умова виконується змінній *res* присвоюємо "Cool", інакше присвоюємо *res* значення "Nice weather we're having". Виводимо змінну *res* на екран.

### Блок-схема:



### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
0	A cold, isn't it?
1	Cool
10	Nice weather we're having

## Задача 7.

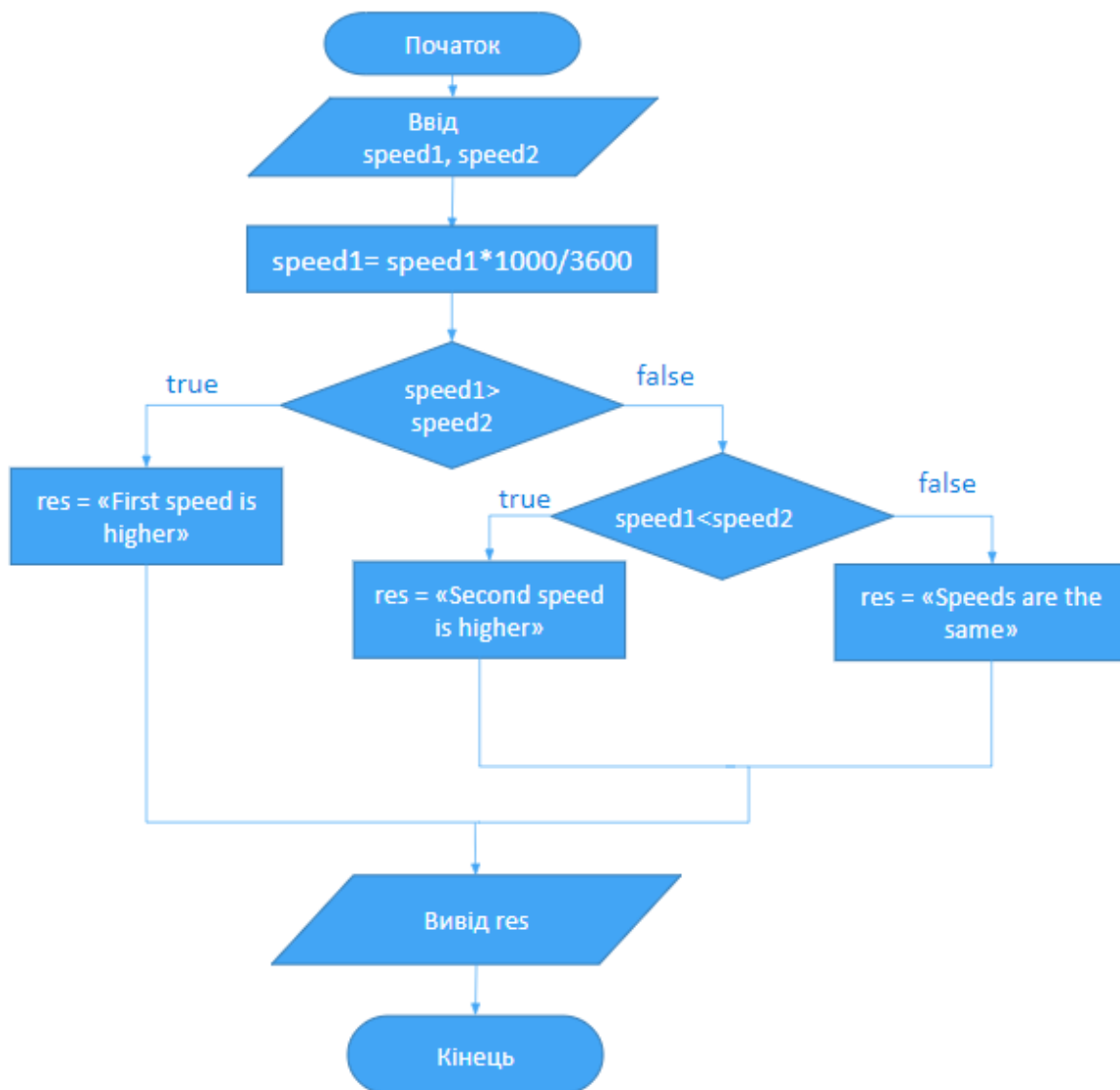
Відомі дві швидкості: одна задана в кілометрах за годину, інша - в метрах за секунду. Яка з швидкостей більша?

### Пояснення розв'язку

Вводимо два числа: швидкість в кілометрах за годину *speed1* і швидкість в метрах за секунду *speed2*. Перед порівнянням переведемо *speed1* в метри за секунду. Для цього використаємо формулу  $speed1 = speed1 * 1000 / 3600$  (так як 1 км = 1000 м і 1 год. = 3600 с). Далі порівнюємо вже змінене значення *speed1* зі *speed2*. Якщо  $speed1 > speed2$  змінній *res* присвоюємо значення "First speed is higher", інакше перевіряємо умову  $speed1 < speed2$ . Якщо ця умова виконується, то *res* присвоюємо значення "Second speed is higher", інакше *res* присвоюємо значення "Speeds are the same". Виводимо *res* на екран.

### Блок-схема:





#### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10 20	Second speed is higher
20 5	First speed is higher
36 10	Speeds are the same

## Задача 8.

Напишіть програму, яка запитує два цілих числа. Якщо добуток чисел перевищує їх суму, надрукувати добуток чисел, у протилежному випадку - вивести їх суму. Якщо ж добуток дорівнює сумі, вивести різницю чисел.

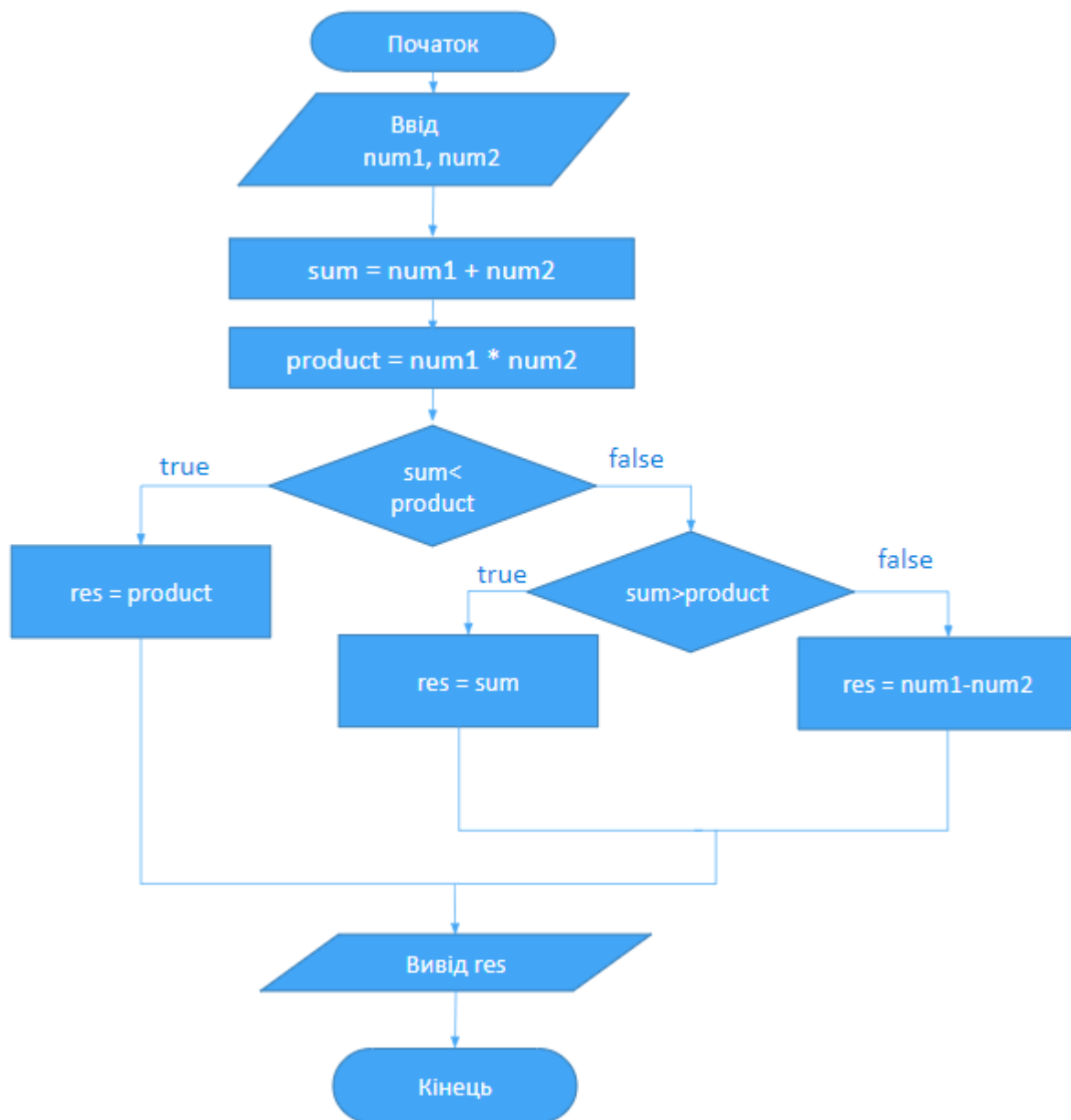
### Пояснення розв'язку

*Ввести два числа  $num1$  та  $num2$ . Знаходимо їх суму та записуємо в комірку  $sum$ .*

*Знаходимо їх добуток та записуємо в комірку  $product$ . Порівнюємо  $sum < product$ , якщо так, то  $res$  присвоюємо  $product$ . Якщо  $sum > product$ , то  $res$  присвоюємо значення  $sum$ . У випадку рівності  $sum$  і  $product$ ,  $res$  присвоюємо  $num1 - num2$ .*

*Виводимо  $res$ .*

### Блок-схема:



#### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
10 20	200
1 2	3
2 2	0

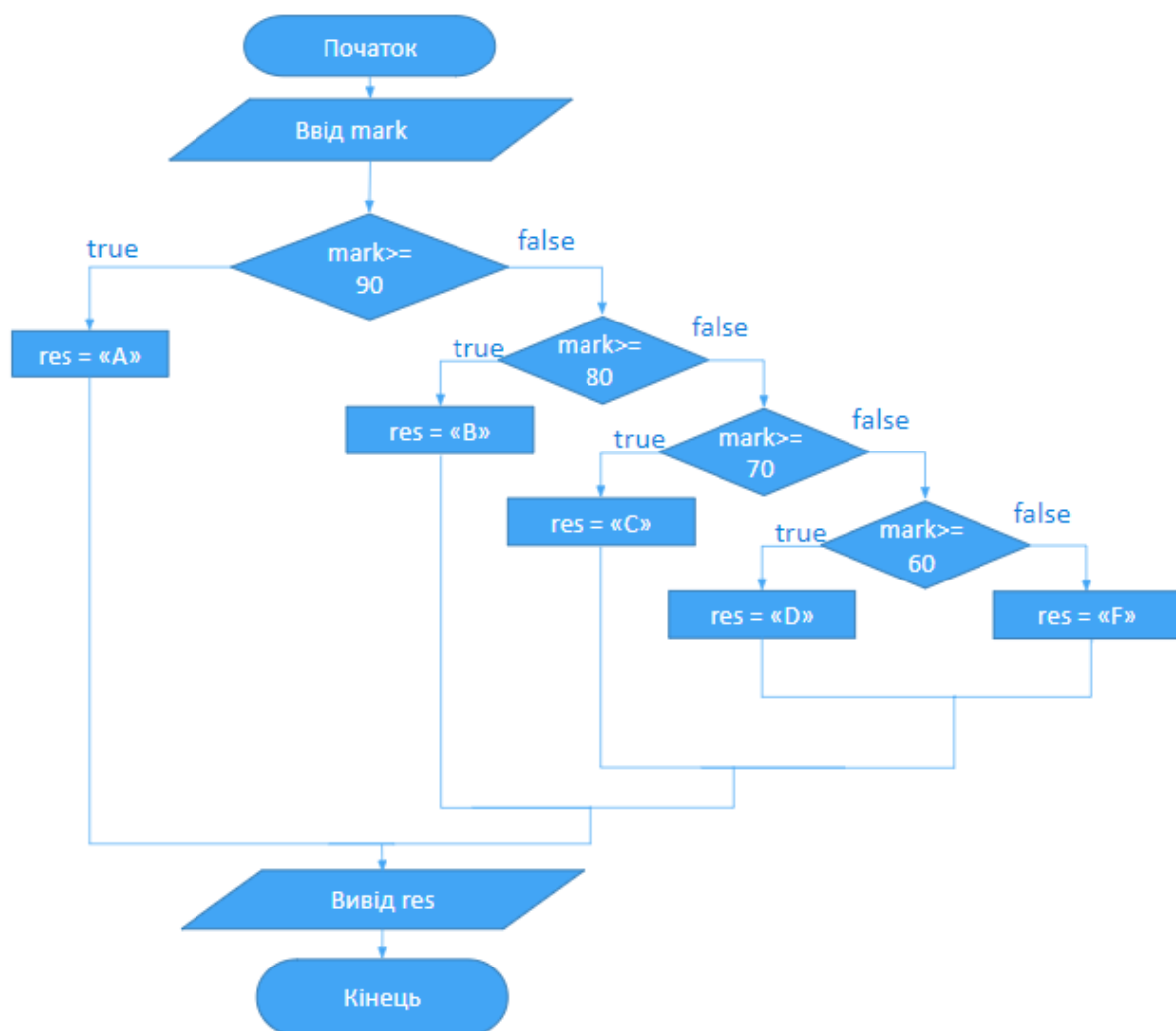
## Задача 9.

В університеті використовується наступна шкала для інтерпретації результатів тестування студентів: 90 балів і вище – оцінка А, 80-89 балів – оцінка В, 70-79 балів – оцінка С, 60-69 балів – оцінка D, нижче 60 – оцінка F. Напишіть програму, яка на основі введеного тестового балу повертає оцінку для цього балу.

### Пояснення розв'язку

Ввести значення оцінки *mark*. Порівнюємо *mark* з 90, якщо  $mark \geq 90$ , то *res* присвоюємо "А", далі перевіряємо  $mark \geq 80$ , то *res* присвоюємо "В". Наступне порівняння  $mark \geq 70$ , і *res* = "С", далі  $mark \geq 60$ , то *res*="D", і нарешті, *res*="F". Виводимо *res* на екран.

### Блок-схема:



### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
90	A
80	B

79	C
60	D
59	F

## Задача 10\*.

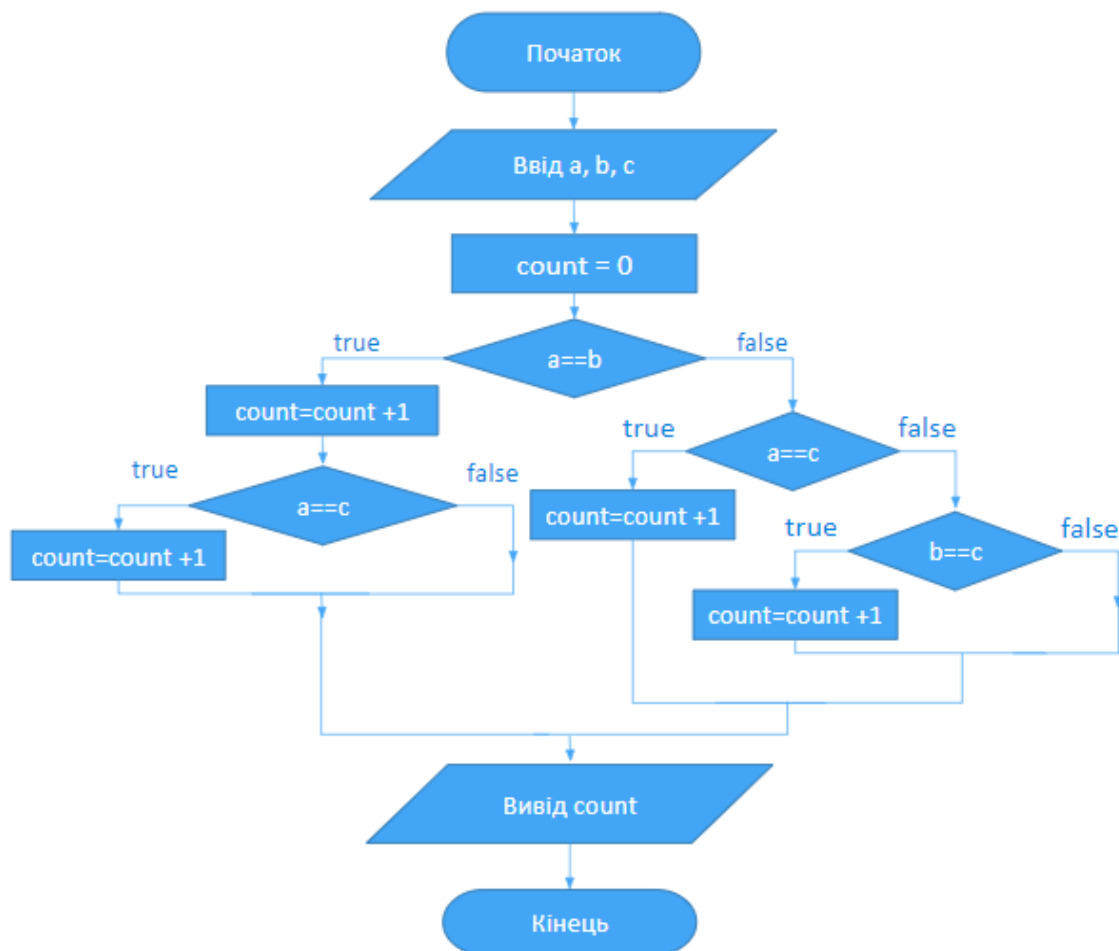
Дано три цілих числа. Визначте, скільки серед них збігаються. Програма повинна вивести одне з чисел: 2 – якщо усі числа однакові, 1 – якщо два числа однакові, або 0 – якщо усі введені числа різні.

### Пояснення розв'язку

Вводимо три числа  $a$ ,  $b$ ,  $c$ . Виберемо змінну  $count$  для результату. Початково присвоємо їй значення 0. Порівнюємо на рівність  $a$  і  $b$ . Якщо виявиться що вони рівні, то  $count$  присвоюємо  $count + 1$ . Далі порівнюємо на рівність  $a$  і  $c$ . Якщо вони рівні, то знову  $count = count + 1$ .

Якщо ж при порівнянні  $a$  і  $b$  виявиться, що вони різні, то порівнюємо на рівність  $a$  і  $c$ . Якщо вони рівні, то  $count = count + 1$ , інакше порівнюємо на рівність  $b$  і  $c$ . У випадку рівності  $count = count + 1$ , якщо умова що  $b == c$  не виконується, то ніяких додаткових дій не виконуємо. І нарешті виводимо  $count$ .

**Блок-схема:**



#### Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
1 2 3	0
2 3 2	1
5 5 5	2