Розвиваємо алгоритмічне мислення

Алгоритми зі списками

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.

Створити список, який містить 10 значень цілого типу. Організувати введення цього списку з клавіатури. Вивести отриманий список на екран. Знайти суму всіх елементів списку.

Пояснення розв'язку

Оголошуємо список `mas`*.*

Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 9 з кроком 1. В тілі циклу організовуємо введення значення цілого типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

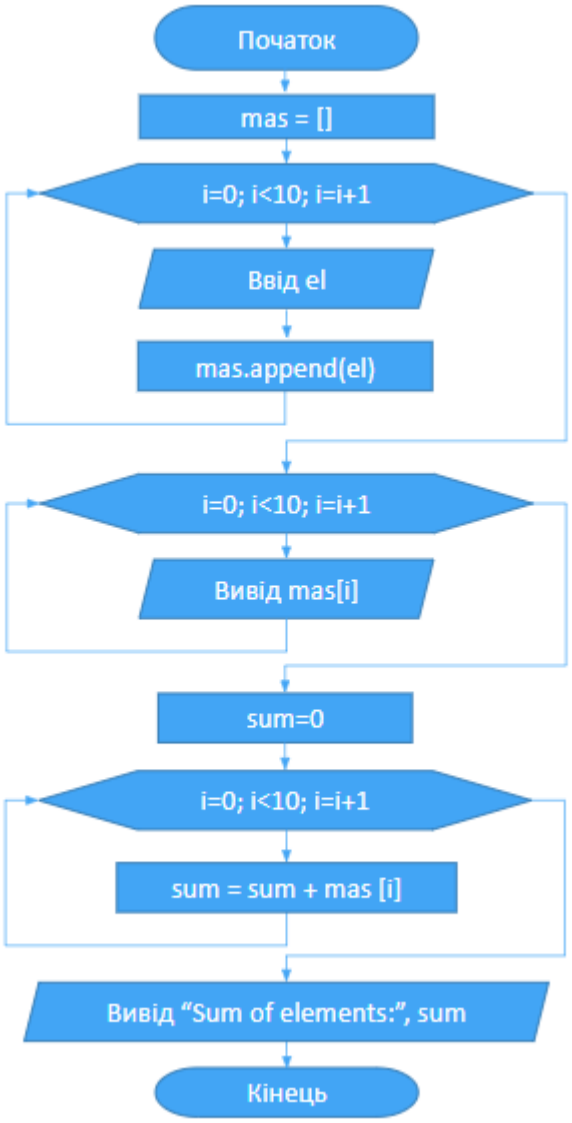
Повторно використовуємо цикл з параметром `i` *та організовуємо вивід елементів списку* `mas` *на екран.*

Оголошуємо змінну `sum` *для знаходження суми та присвоюємо їй значення 0.*

Втретє використовуємо цикл з параметром `i` *та почергово додаємо елементи списку* `mas` *до значення змінної* `sum`*.*

Виводимо значення `sum` *на екран.*

Блок-схема:



Код:

```
mas = []
for i in range(10):
    el= int(input())
    mas.append(el)
for i in range(10):
    print(mas[i], end=" ")
sum = 0
for i in range(10):
    sum= sum + mas[i]
print()
print("Sum of elements: ",sum)
```

Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 Sum of elements: 10
1 2 3 6 5 4 9 8 7 6	1 2 3 6 5 4 9 8 7 6 Sum of elements: 51

Задача 2.

Створити список , який складатиметься з 7 значень дійсного типу. Організувати введення цього списку з клавіатури. Знайти добуток додатних елементів списку.

Пояснення розв'язку

Оголошуємо список `mas`*.*

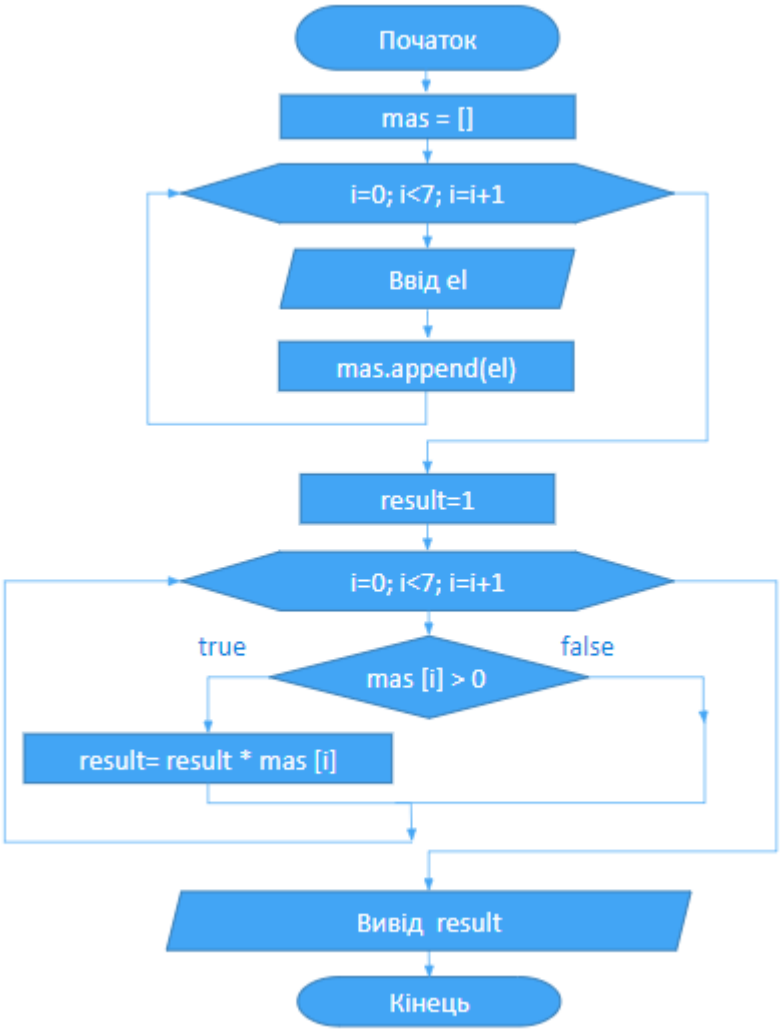
Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 6 з кроком 1. В тілі циклу організовуємо введення значення дійсного типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

Оголошуємо змінну `result` *для знаходження добутку додатних елементів та присвоюємо їй значення 1.*

Повторно використовуємо цикл з параметром `i`*. В циклі перевіряємо чи чергове значення списку* `mas` *додатне, якщо так то, домножаємо значення змінної* `result` *на відповідне значення списку* `mas` *.*

Виводимо значення `result` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
1.2 -4 5.8 -6.2 8.4 1.2 -7	70.15679999999999
2 1 -7 -4.5 -8 3.5 -9	7.0

Задача 3.

Створити список , який складатиметься з 9 значень цілого типу. Організувати введення цього списку з клавіатури. Знайти кількість від’ємних елементів списку

Пояснення розв’язку

Оголошуємо список `mas`*.*

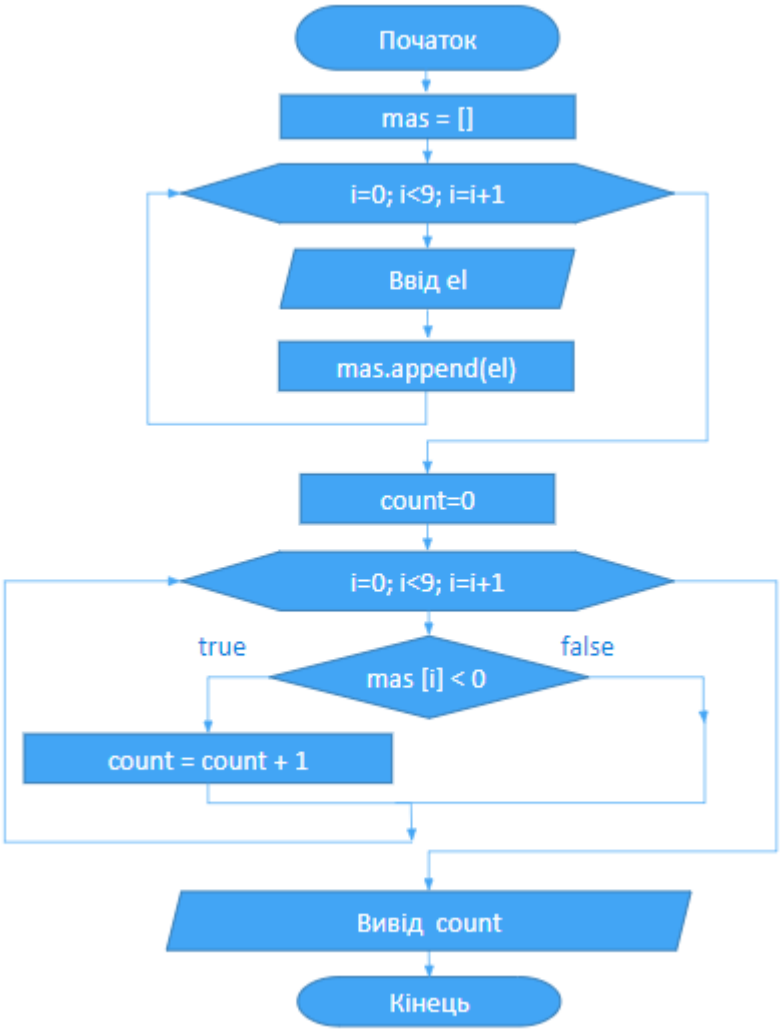
Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 8 з кроком 1. В тілі циклу організовуємо введення значення цілого типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

Оголошуємо змінну `count` *для знаходження кількості від'ємних елементів та присвоюємо їй значення 0.*

Повторно використовуємо цикл з параметром `i`*. В циклі перевіряємо чи чергове значення списку* `mas` *від'ємне, якщо так, то збільшуємо значення змінної* `count` *на 1.*

Виводимо значення `count` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
4 -7 8 -7 1 2 -1 7 -2	4
1 1 1 1 1 1 1 1 1	0

Задача 4.

Створити список , який складатиметься з 6 значень дійсного типу. Організувати введення цього списку з клавіатури. Знайти максимальний елемент списку

Пояснення розв'язку

Оголошуємо список `mas`*.*

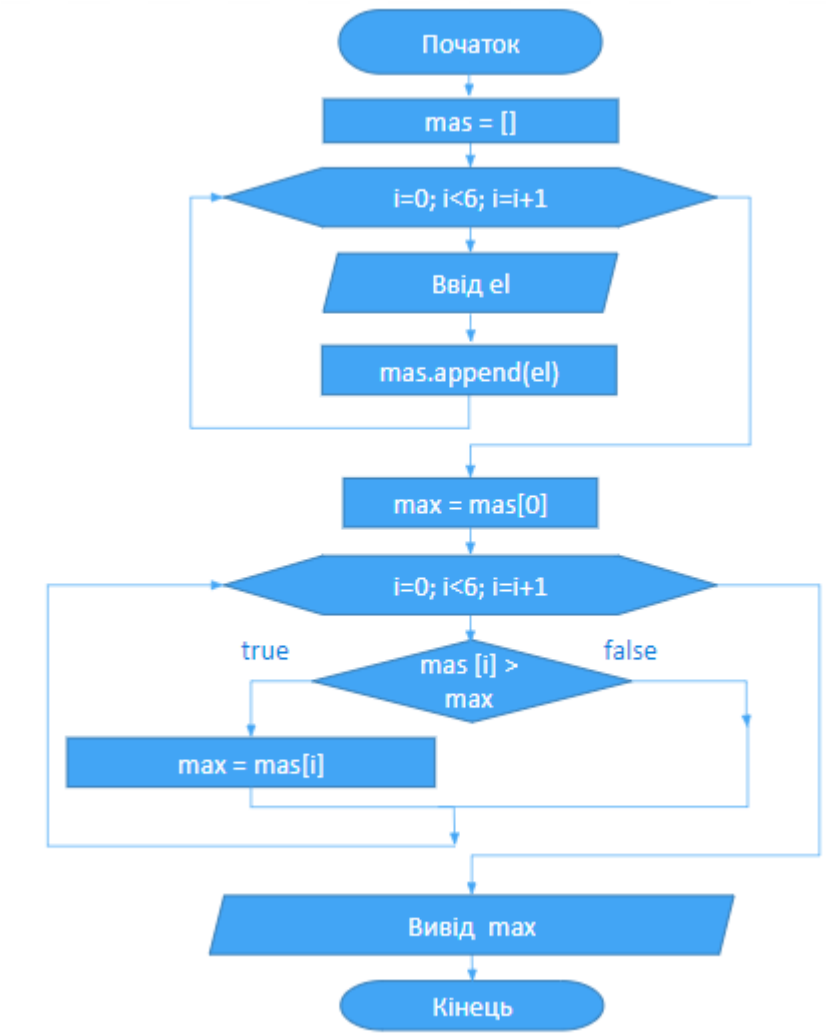
Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 5 з кроком 1. В тілі циклу організовуємо введення значення дійсного типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

Оголошуємо змінну `max` *для знаходження максимального елемента списку та присвоюємо їй значення першого елемента списку (списку)* `mas [0]`*.*

Повторно використовуємо цикл з параметром `i`*. В циклі перевіряємо чи чергове значення списку* `mas` *є більшим за* `max`*, якщо так, то присвоюємо це значення змінній* `max`*.*

Виводимо значення `max` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
4 3.8 8.5 -9 6.7 7.1	8.5
10.2 9.1 -14 4 1 5	10.2

Задача 5.

Створити список , який складатиметься з 12 значень дійсного типу. Організувати введення цього списку з клавіатури. Знайдіть кількість елементів списку (включаючи максимальний), які відмінні від найбільшого елемента не більше ніж на 30%.

Пояснення розв’язку

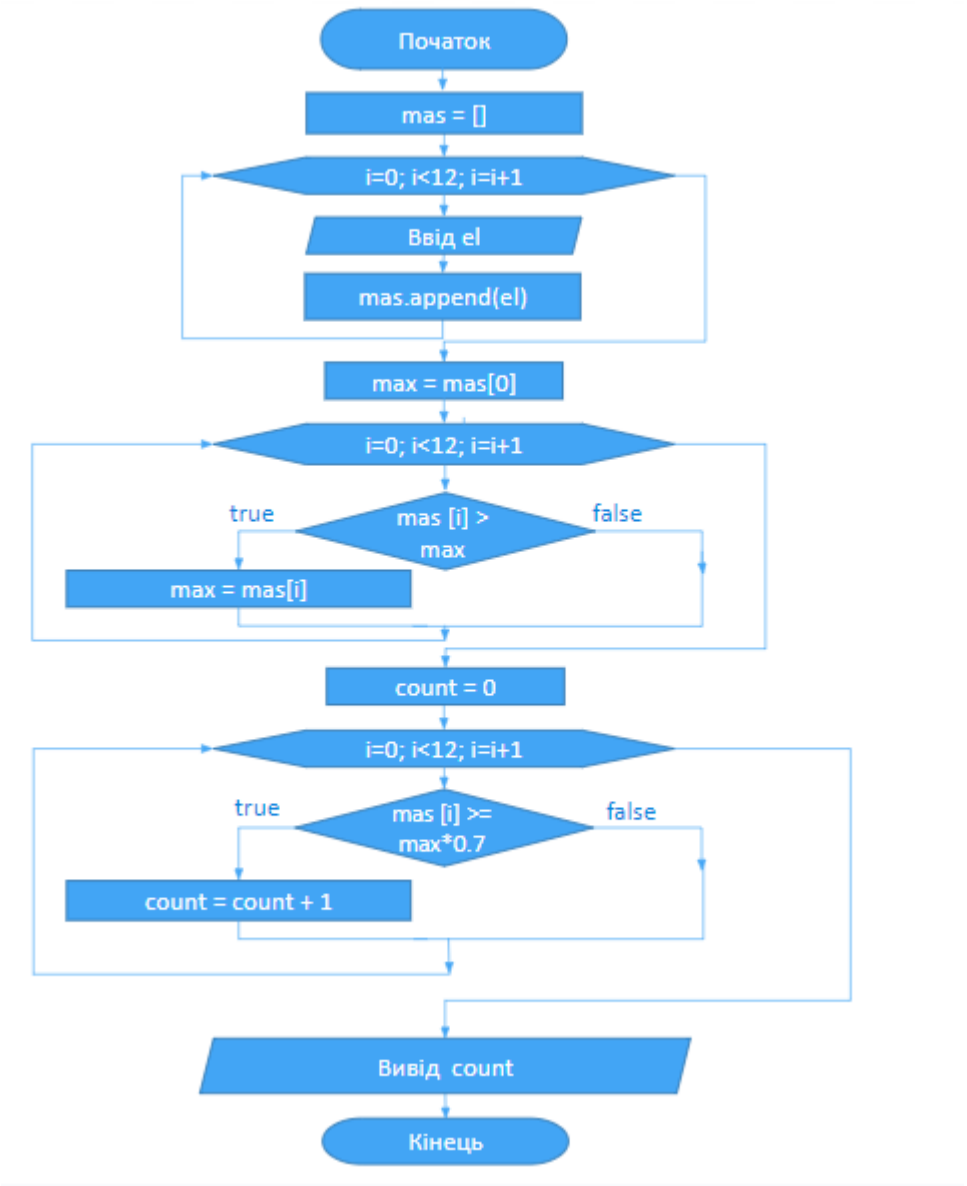
Для розв'язку задачі 5 скористаємося розв'язком задачі 4 (введення елементів списку та пошук максимального).

Тепер оголошуємо змінну `count`, *для знаходження кількості елементів, що відрізняються від максимального не більше ніж на 30% та присвоюємо їй значення 0.*

Втретє використовуємо цикл з параметром `i` *та почергово порівнюємо чи значення елементів списку* `mas` *більші або рівні значенню* `max*0.7`*, якщо так, то збільшуємо значення змінної* `count` *на 1.*

Виводимо значення `count` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
2 8 15.6 -1 10 12 13.4 -8 12.4 2 -8 1.5	4
7 5	4

9 24 18 -14.5 19.5 8 16.8 12.3 -1.2 7	

Задача 6.

Створити список, який складатиметься з 8 значень дійсного типу. Організувати введення цього списку з клавіатури. Поміняти місцями максимальний та мінімальний елементи списку.

Пояснення розв'язку

Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 7 з кроком 1. В тілі циклу організовуємо введення значення дійсного типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

Оголошуємо змінну `max` *для знаходження максимального елемента списку та присвоюємо їй початкове значення* `mas[0]`*.*

Оголошуємо змінну `min` *для знаходження мінімального елемента списку та присвоюємо їй початкове значення* `mas[0]`*.*

Оголошуємо ще дві змінні `imax` *та* `imin` *для збереження відповідно індексів максимального та мінімального елементів списку та присвоюємо їм значення 0.*

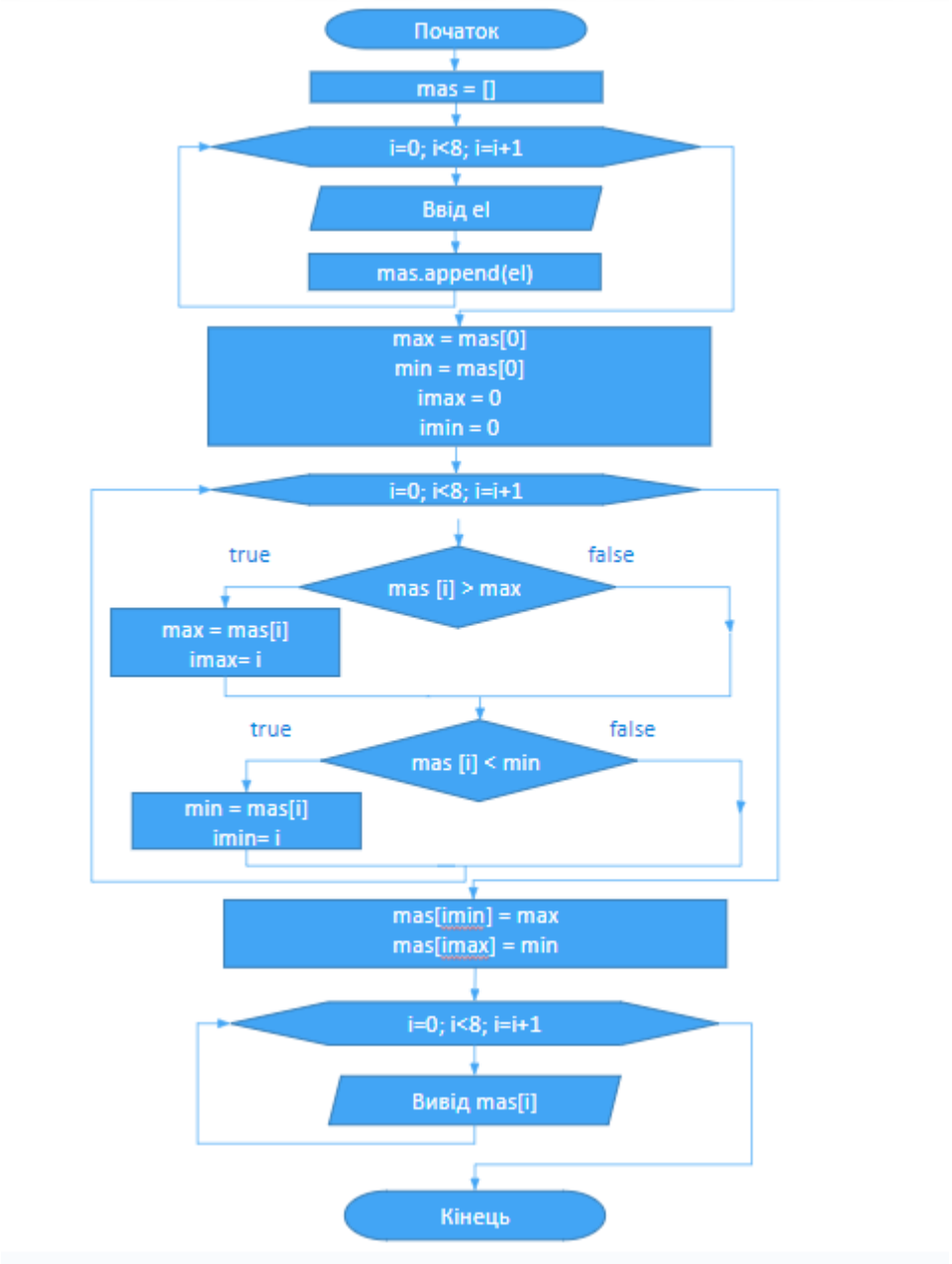
Повторно використовуємо цикл з параметром `i`*. В циклі перевіряємо чи чергове значення списку* `mas` *є більшим за* `max`*, якщо так, то присвоюємо це значення змінній* `max` *і при цьому змінній* `imax` *присвоюємо значення рівне* `i`*. Також перевіряємо чи чергове значення списку* `mas` *є меншим за* `min`*, якщо так, то присвоюємо це значення змінній* `min` *і при цьому змінній* `imin` *присвоюємо значення рівне* `i`*.*

Після завершення циклу робимо заміну `mas[imax]= min`*, а* `mas[imin]=max`*.*

Втретє використовуємо цикл з параметром `i` *та організовуємо вивід елементів списку* `mas` *на екран.*

.

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
2.4 8 7 -1.5 4.2 9 -6.3 11	2.4 8 7 -1.5 4.2 9 11 -6.3
1.4 -7 1.3 8.2	1.4 14 1.3 8.2

14 -1 -2 8	-7 -1 -2 8

Задача 7.

Створити список, який складатиметься з 10 значень цілого типу. Організувати введення цього списку з клавіатури. Серед елементів з непарними номерами (індексами) знайдіть найбільший елемент списку, який ділиться на 3 без остачі. (Гарантовано, що в списокі є хоча б один непарний елемент, який ділиться на 3).

Пояснення розв'язку

Оголошуємо список `mas`*.*

Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до 9 з кроком 1. В тілі циклу організовуємо введення значення цілого типу у змінну* `el`*. Значення змінної* `el` *заносимо в список* `mas`*.*

Оголошуємо змінну `ind` *та присвоюємо їй значення 1.*

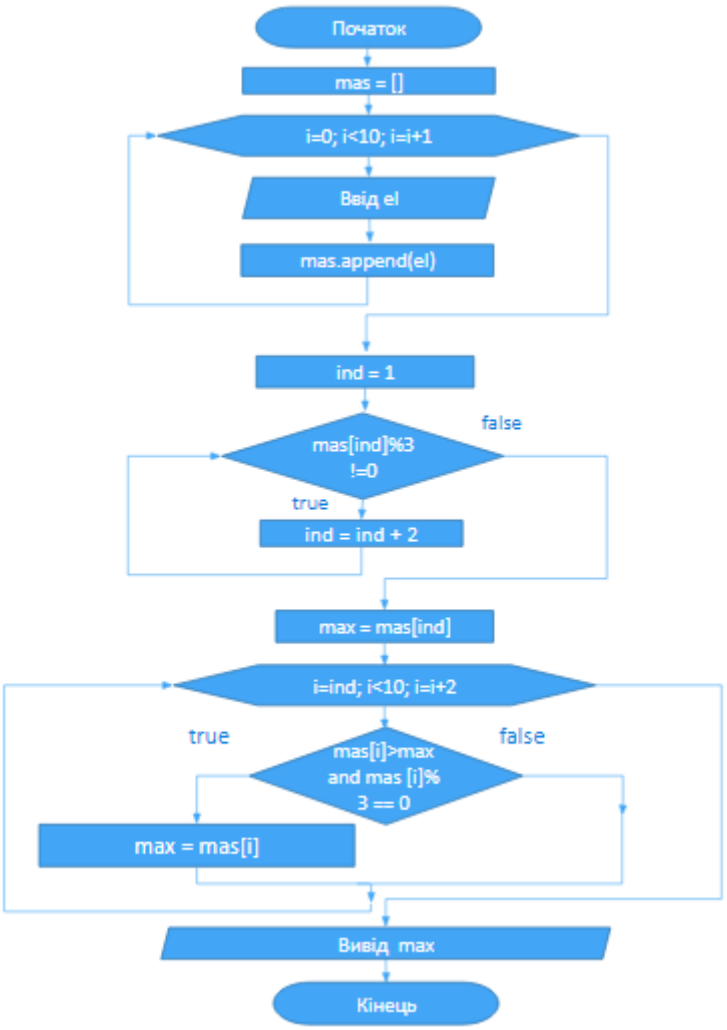
Організовуємо цикл `while` *з умовою* `mas[ind]%3 !=0`*. В тілі циклу збільшуємо значення* `ind` *на 2. В цьому циклі ми знайдемо перший елемент списку з непарним номером, який кратний 3.*

Оголошуємо змінну `max` *для знаходження максимального елемента списку та присвоюємо їй значення* `mas[ind]`*.*

Повторно використовуємо цикл з параметром `i`*, який змінюється від* `ind` *до 9 з кроком 2. В циклі перевіряємо чи виконуються одночасно дві умови* `mas[i] > max` *та* `mas[i]%3 == 0`*, якщо так, змінній* `max` *присвоюємо значення mas[i].*

Виводимо значення `max` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
3 4 5 6 7 8 9 1 2 7	6
24 -1 48 -9 23 -6 9 0	0

-8 -74	

Задача 8.

З клавіатури вводять число. Знайти суму цифр числа (див. задачу 9 алгоритми з циклами)

Пояснення розв'язку

Організовуємо ввід числа `n` *з клавіатури.*

Оголошуємо список `mas`*.*

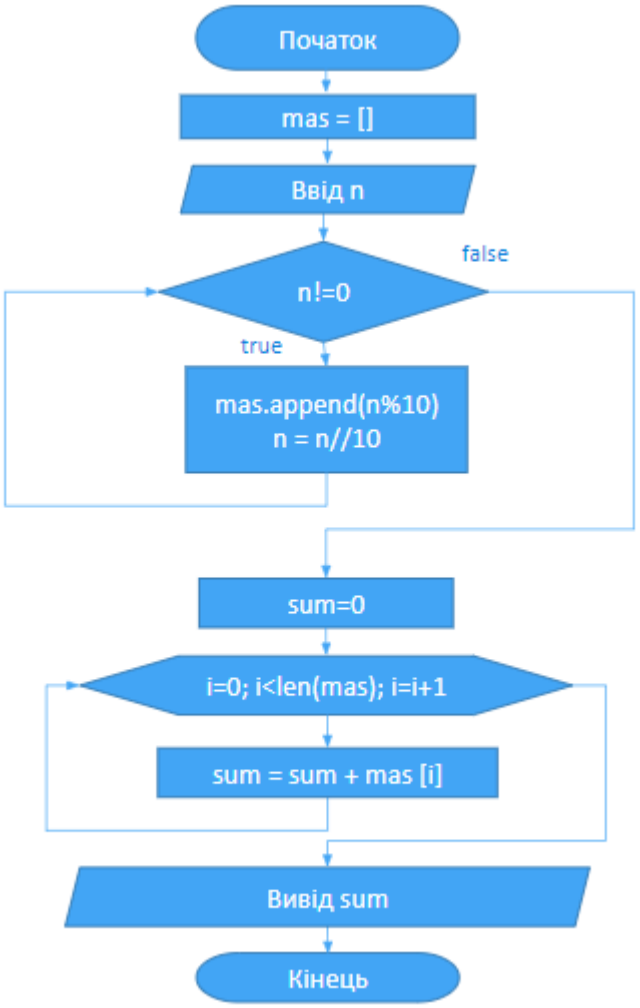
Використовуючи цикл while з умовою n !=0. В тілі циклу в список `mas` *додаємо значення* `n%10` *та виконуємо дію* `n=n//10`*. Після закінчення циклу в списокі* `mas` *буде записано всі цифри числа* `n`*.*

Оголошуємо змінну `sum` *для знаходження суми та присвоюємо їй значення 0.*

Використовуємо цикл з параметром `i`*, який змінюється від 0 до* `len(mas)-1` *з кроком 1 та почергово додаємо елементи списку* `mas` *до значення змінної* `sum`*.*

Виводимо значення `sum` *на екран.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
123	6
400006	10

Задача 9.

Напишіть програму, яка формує список з 10 цілих чисел - членів арифметичної прогресії (користувач задає значення першого члена і різницю арифметичної прогресії). Знайдіть суму десяти членів цієї арифметичної прогресії.

Пояснення розв'язку

Арифметична прогресія - це послідовність чисел, кожен член якої, починаючи з другого, утворюється додаванням до попереднього члена одного й того ж числа, яке

називають різницею прогресії. Наприклад, в прогресії 5, 8, 11, 14, 17, 20, 23, … перший член прогресії – 5, різниця – 3.

Оголошуємо список `mas`.

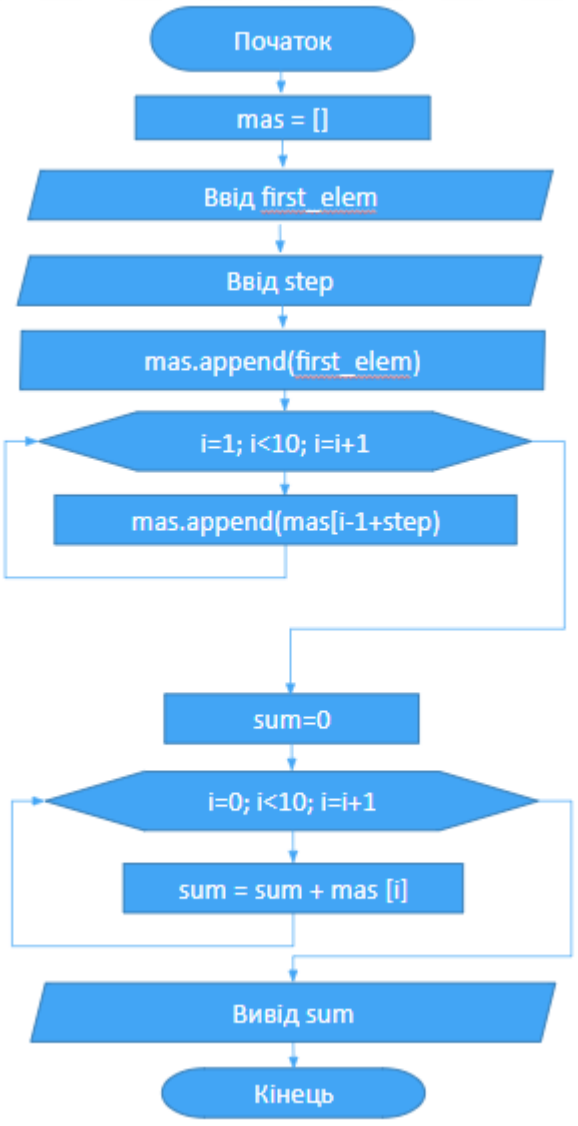
Організовуємо ввід змінних `first_elem` *та* `step` *з клавіатури.*

Додаємо до списку `mas` *значення* `first_elem`.

Організовуємо цикл з параметром `i`, *де* `i` *буде змінюватися від 1 до 9 з кроком 1. В тілі циклу організовуємо введення списку за формулою* `mas.append(mas[i-1]+step)`.

Далі для підрахунку суми членів прогресії скористаємося розв'язком задачі 1 для знаходження суми елементів списку та виводимо результат на екран.

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
1 1	55

2 -3	-115
2 3	155

Задача 10.

Перевірити чи введене число є паліндромом

Пояснення розв’язку

Паліндром *(перевертень) — слово, число, набір символів, словосполучення або віршований рядок, що однаково читається в обох напрямках. Наприклад: 123321, “дід”, “козак з казок”.*

Організовуємо ввід числа `n` *з клавіатури.*

Оголошуємо список `mas`*.*

Використовуючи розв'язок задачі 8, заповнюємо список `mas` *цифрами числа* `n`*.*

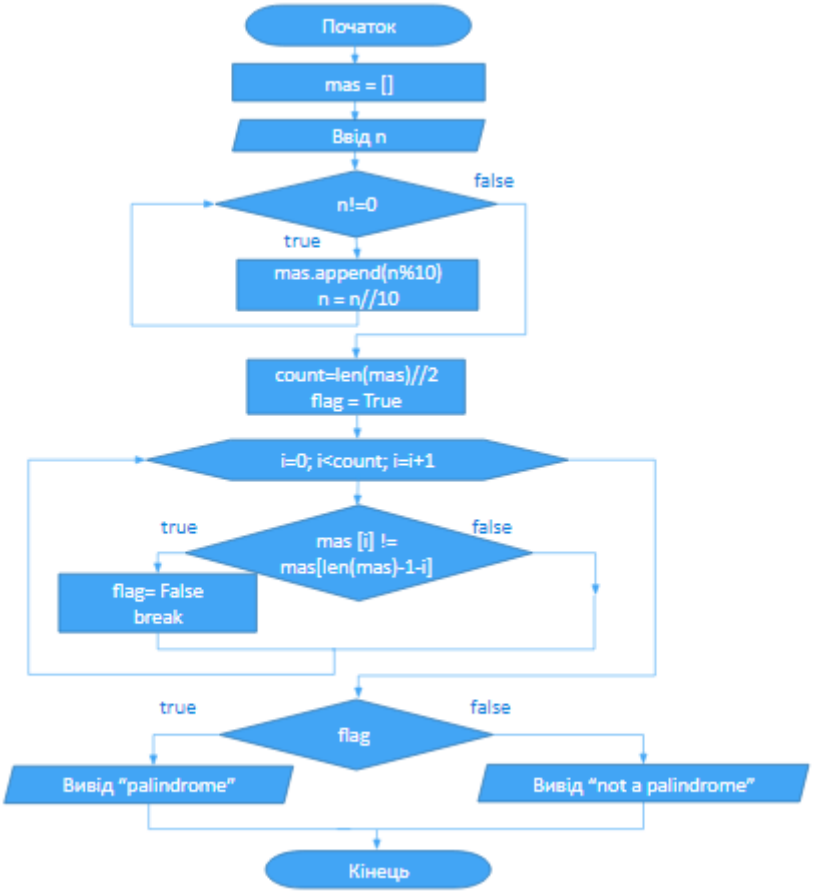
Далі в змінну `count` *збережемо половину довжини списку* `len(mas)//2`*.*

Оголосимо змінну `flag`*, якій присвоємо початкове значення* ***True****. Ця змінна буде індикатором того, що відповідні числа співпадають.*

Організовуємо цикл з параметром `i`*, де* `i` *буде змінюватися від 0 до* `count` *з кроком 1. В тілі циклу організовуємо порівняння mas[i] != mas[len(mas)-1-i], якщо умова виконується то присвоюємо* `flag` *значення* ***`False`*** *і виходимо з циклу.*

Далі перевіряємо значення `flag`*, якщо воно рівне* ***True****, то виводимо на екран “*palindrome*”, інакше виводимо “*not a palindrome*”.*

Блок-схема:



Приклади вхідних даних та результатів

Вхідні дані	Вихідні дані
123454321	palindrome
12354321	not a palindrome
123321	palindrome
4	palindrome