



Become QA Auto

Вказівка повторення. Цикл з параметром

Бутенко Сергій

```
le.py > ...
```

```
= "PiMyLifeUp" #First assign the 'x' variable a string
```

```
type(x) #Output the data type of our 'x' variable
```

```
= 43 #Now assign it a integer value
```

```
type(x) #Output its type again
```

```
days = 5
```

```
print("There are " + str(days) + " days r
```

```
b, c = "Apple", "Banana", "Cherry"
```

```
b = c = "Lime"
```

```
= "Hello World"
```

```
= 453
```

```
int(x)
```

```
int(y)
```

```
= "Life Up"
```





План лекції



Класифікація циклів



Блок-схема задачі про зернини на шаховій дошці



Основні принципи алгоритмів повторення



Дії комп'ютера при виконання циклу з параметром



Цикл з параметром на блок-схемах



Програмна реалізація



Цикли – повторення певних дій



Наперед відома кількість
повторень - обертів



Повторення до настання певної
умови (події) –
додаємо солі по смаку



⦿ Вказівка повторення: класифікація і основні принципи



Цикли з параметром - повторюють тіло циклу наперед відому кількість разів

Цикли «поки» - повторюють тіло циклу поки виконується якась умова

Тіло циклу - дії, які повторюються щоразу при виконання циклу.

Параметр циклу - змінна, від значення якої залежить, чи буде виконуватися тіло циклу.

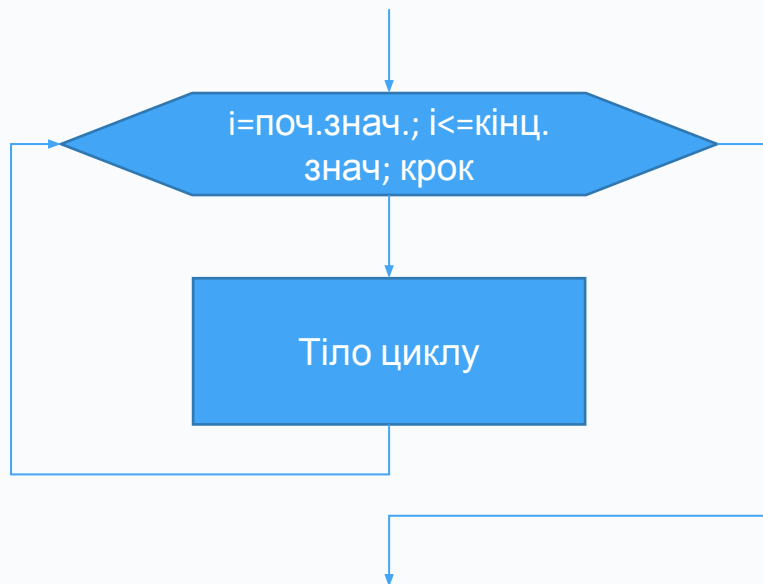
Алгоритми з повторенням повинні підкорятися трьом важливим принципам:

1. Циклічний процес повторює (ітерує) певний підпроцес – тіло циклу.
2. Кожна ітерація повинна змінювати принаймні одне значення (параметр циклу)
3. Повинна бути якась умова, за якої цикл завершується. І система в циклі повинна досягати цього стану.

⦿ Цикл з параметром



Для **параметру** циклу **i**,
що змінюється від **початкового** до **кінцевого**
значення з заданим **кроком**,
виконувати **тіло циклу**



Дії комп'ютера

1. Параметру циклу "*i*" присвоюється початкове значення.
2. Перевіряється умова виконання тіла циклу. Якщо параметр "*i*" менший або рівний за кінцеве значення, то виконується тіло циклу.
3. Параметр циклу "*i*" автоматично збільшується на крок зміни параметру.
4. Знову перевіряється умова *i* ≤ кінцевого значення. Поки умова істинна, буде виконуватися тіло циклу і повторюватися кроки 3 та 4.
5. Якщо *i* > кінцевого значення, то припиняється виконання циклу.



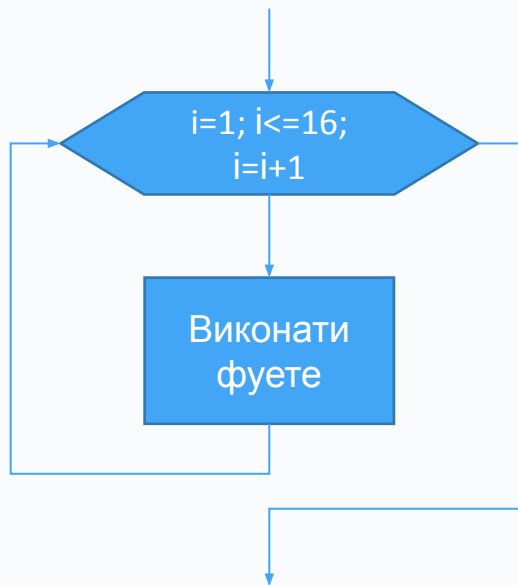
Реалізація циклу з параметром



Для i ,
що змінюється від 1 до 16 з кроком 1,
виконати фуєте



Блок-схема



◎ Задача



Обчислити кількість зерна для винагороди винахідника шахів: за першу клітинку шахівниці видати одну зернину, за другу клітку - видати два зерна, за третю - чотири, за четверту - 8, за п'яту-16, за шосту - 32 ... Нагадаємо, що всього на шахівниці 64 клітинки.



Модель

Дані:

`addition` – черговий доданок, додатне ціле число

Очікуваний результат:

`sum` – сума, додатне ціле число

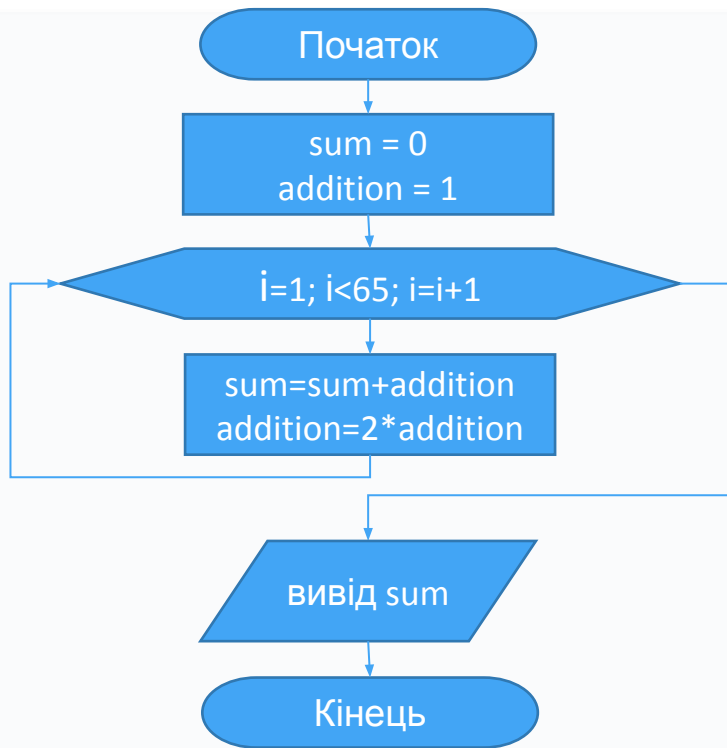
Початкові значення:

`sum=0, addition=1`

Задача зводиться до знаходження суми цілих чисел



Блок-схема алгоритму



Початок – комп'ютер розпочинає виконання алгоритму

Присвоїти початкові значення

сумі `sum = 0`

доданку `addition = 1`

Цикл:

для і від 1 до 64 з кроком 1

виконати тіло циклу:

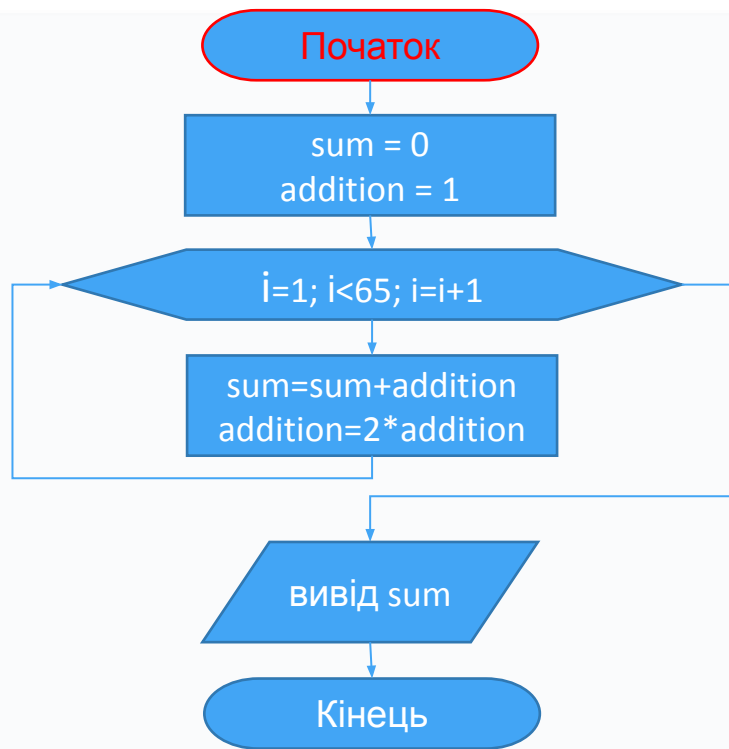
збільшити суму на черговий доданок;
знайти новий доданок, домноживши
попередній на 2

Вивід sum

Кінець



Протокол виконання алгоритму

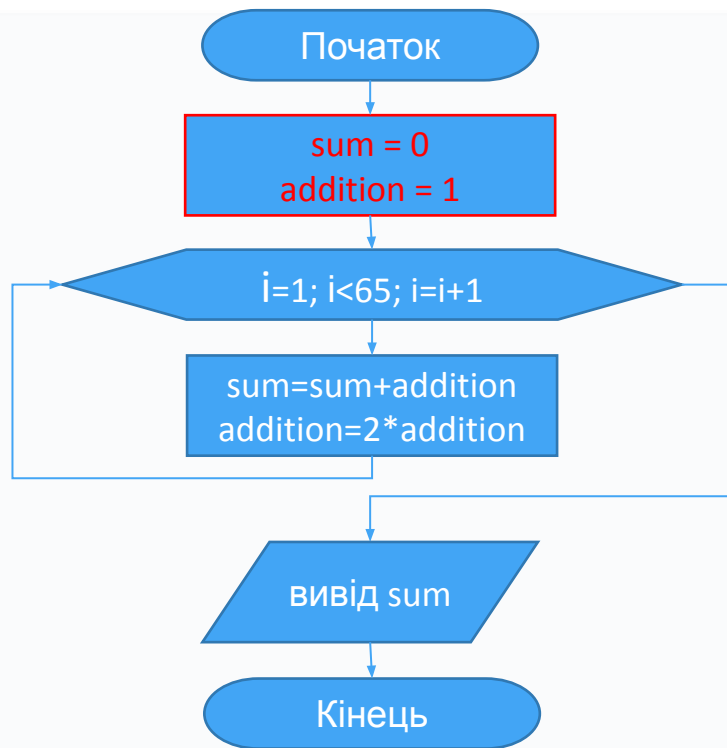


Вміст пам'яті та результат логічних виразів

sum	
addition	
i	
i<65	



Протокол виконання алгоритму

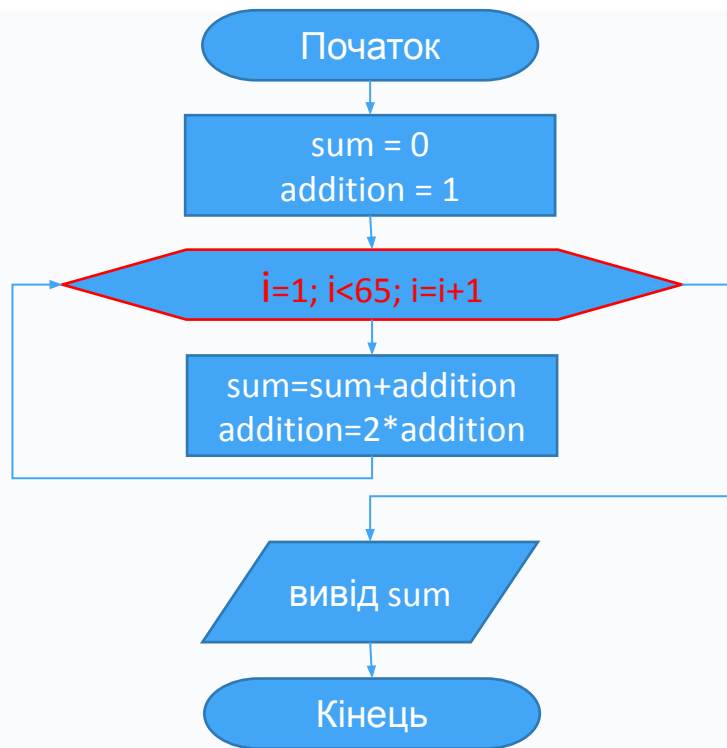


Вміст пам'яті та результат логічних виразів

sum	0
addition	1
i	
i<65	



Протокол виконання алгоритму

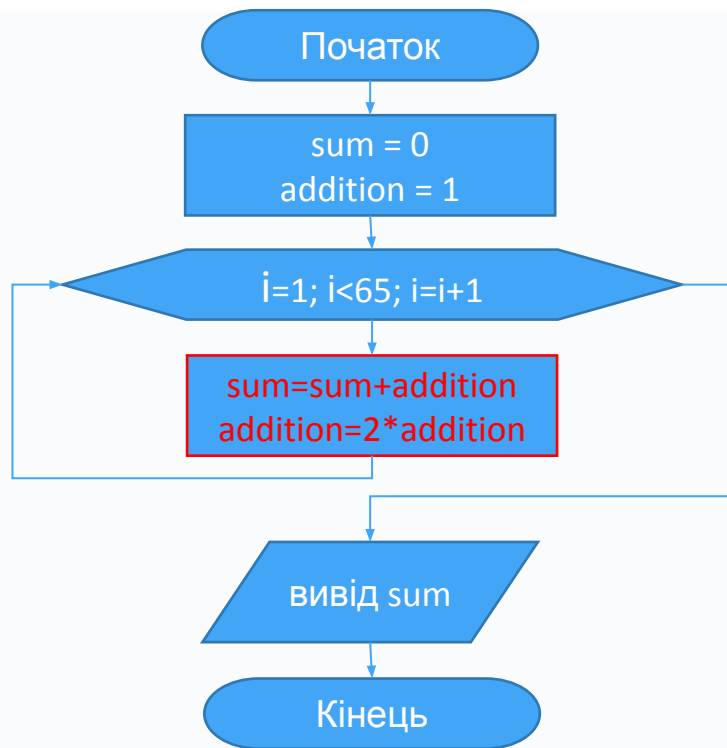


Вміст пам'яті та результат логічних виразів

sum	0
addition	1
i	1
i<65	True



Протокол виконання алгоритму

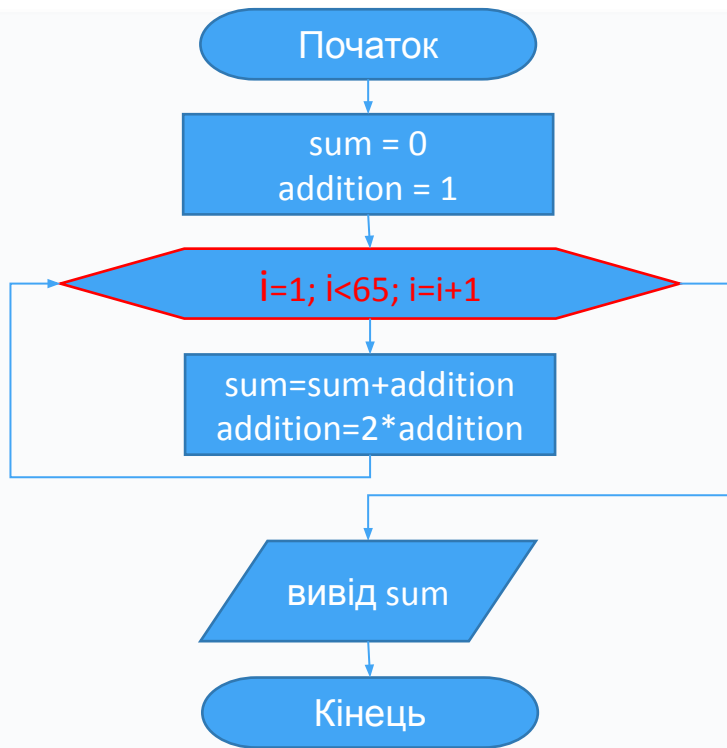


Вміст пам'яті та результат логічних виразів

sum	1
addition	2
i	1
i<65	True



Протокол виконання алгоритму

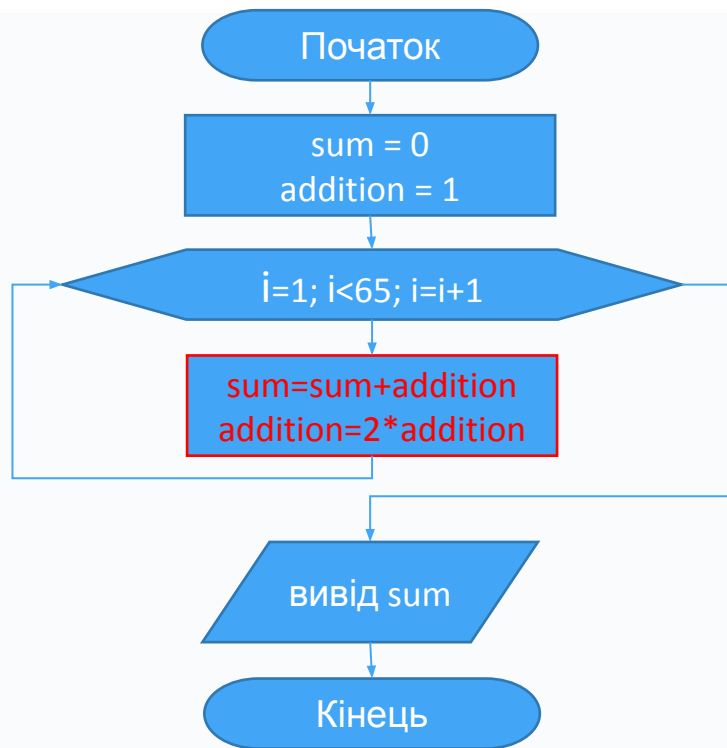


Вміст пам'яті та результат логічних виразів

sum	1
addition	2
i	2
i<65	True



Протокол виконання алгоритму

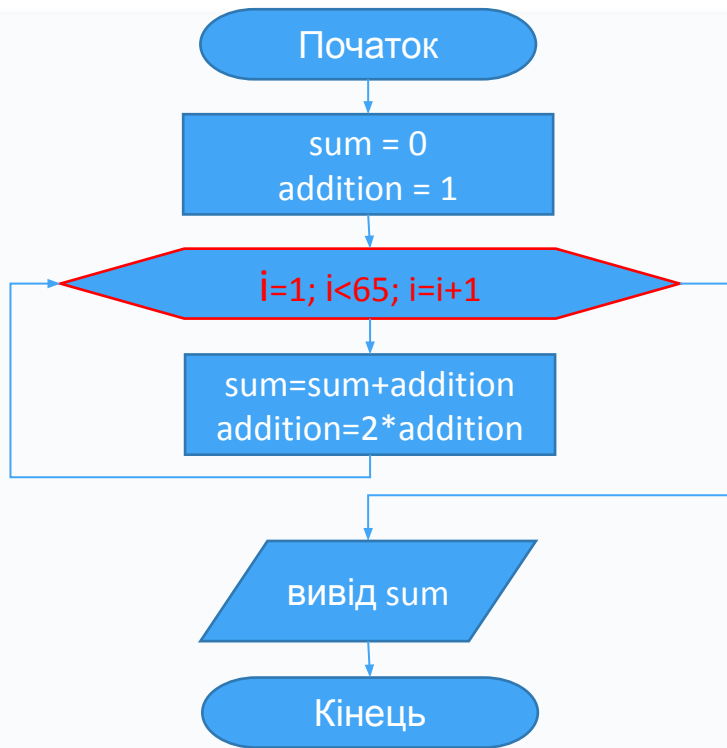


Вміст пам'яті та результат логічних виразів

sum	3
addition	4
i	2
i<65	True



Протокол виконання алгоритму

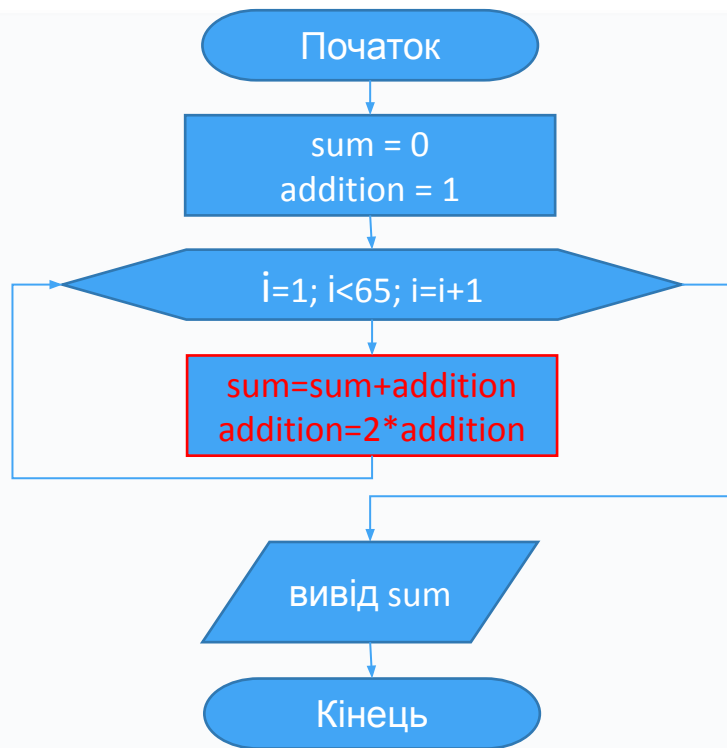


Вміст пам'яті та результат логічних виразів

sum	3
addition	4
i	3
i<65	True



Протокол виконання алгоритму

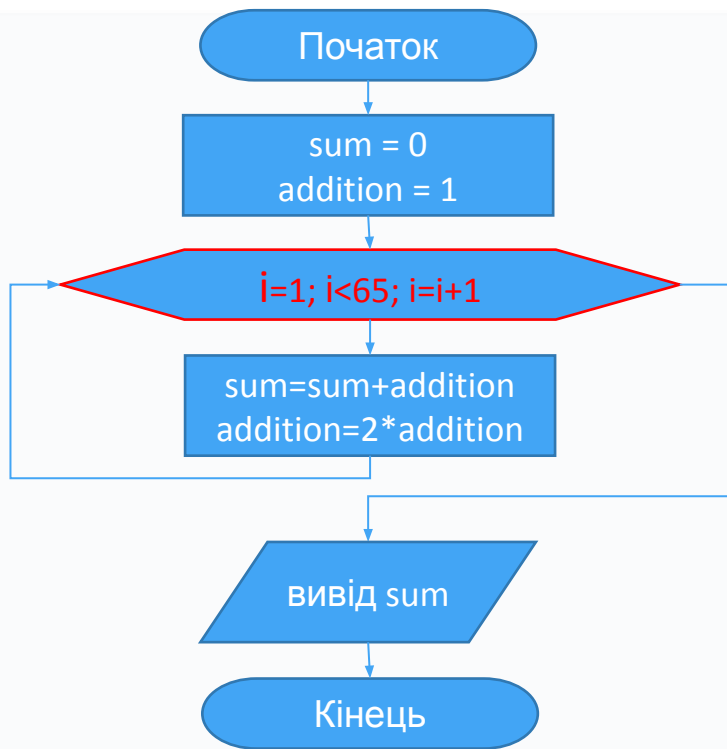


Вміст пам'яті та результат логічних виразів

sum	7
addition	8
i	3
i<65	True



Протокол виконання алгоритму

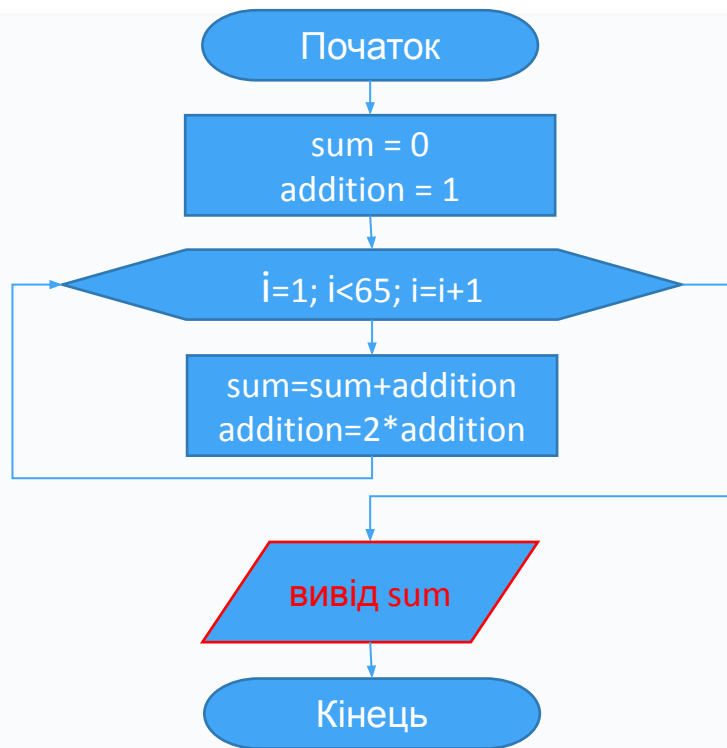


Вміст пам'яті та результат логічних виразів

sum	18446744073709551615
addition	18446744073709551616
i	65
i<65	False



Протокол виконання алгоритму

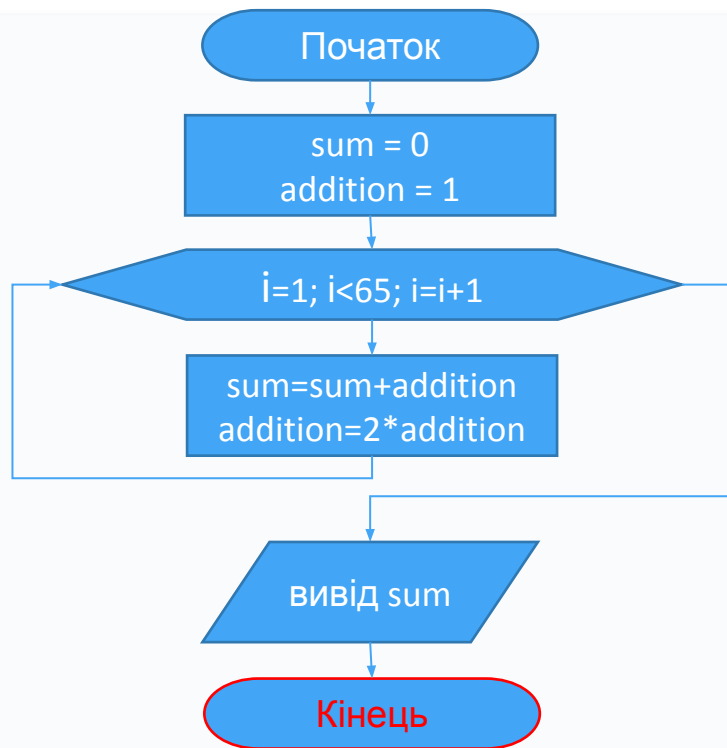


Вміст пам'яті та результат логічних виразів

sum	18446744073709551615
addition	18446744073709551616
i	65
i<65	False



Протокол виконання алгоритму

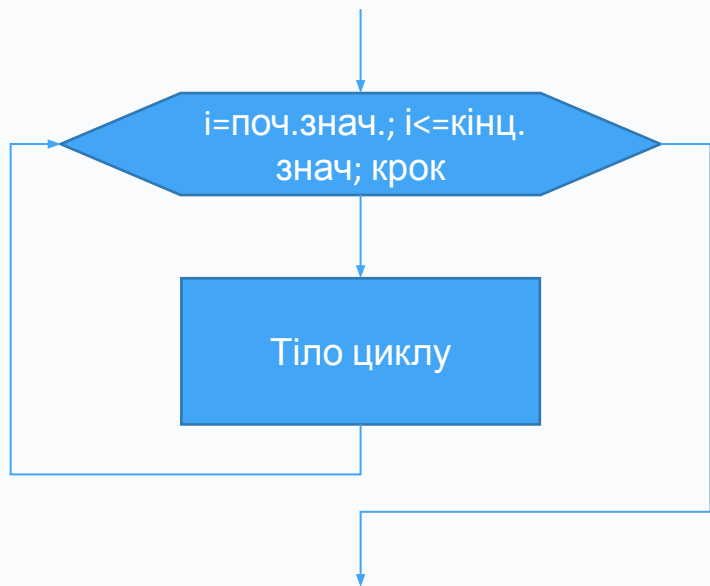


Вміст пам'яті та результат логічних виразів

sum	18446744073709551615
addition	18446744073709551616
i	65
i<65	False



Програмна реалізація циклу з параметром (Python)

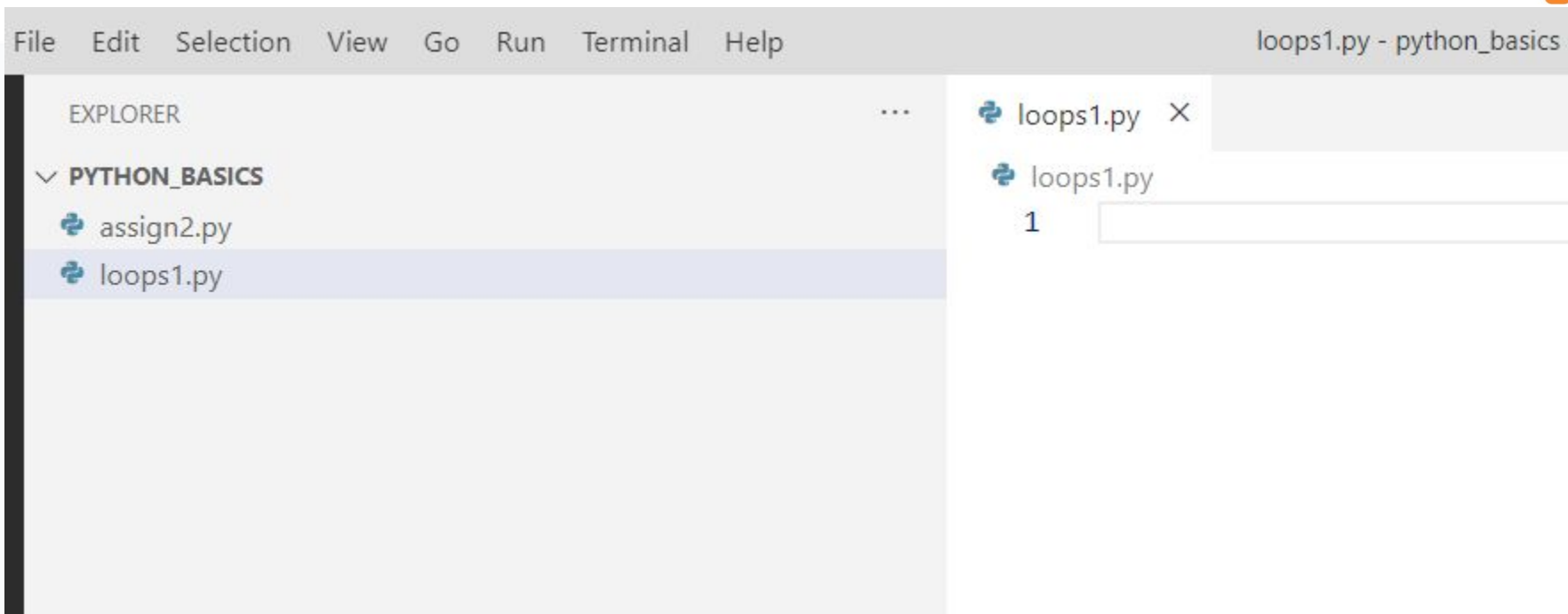


Для **параметру** циклу `i`,
що змінюється від **початкового** до **кінцевого значення** з
заданим **кроком**,
виконувати **тіло циклу**

for **параметр** in range (**поч.знач.**, **кінц.знач.+1**, **крок**):
тіло циклу

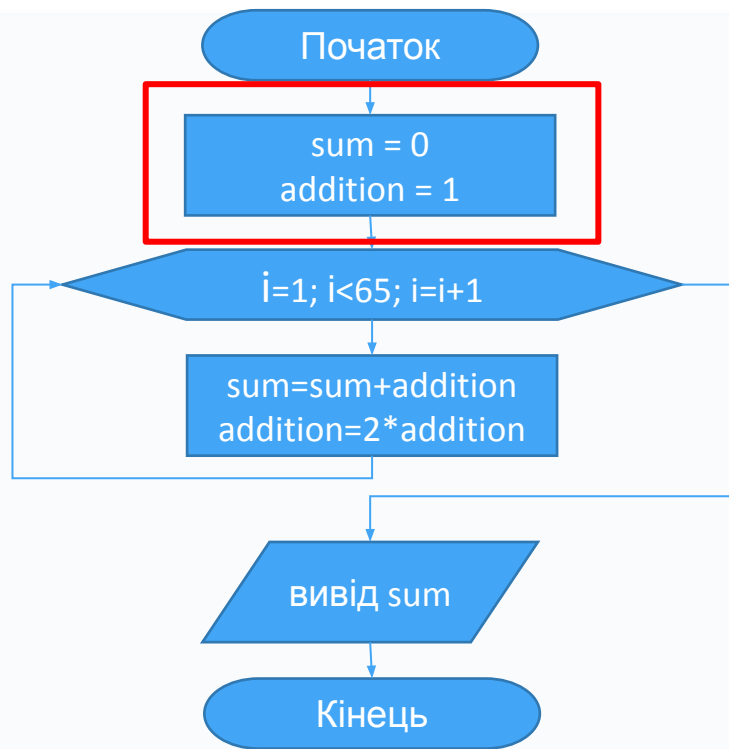


Практика





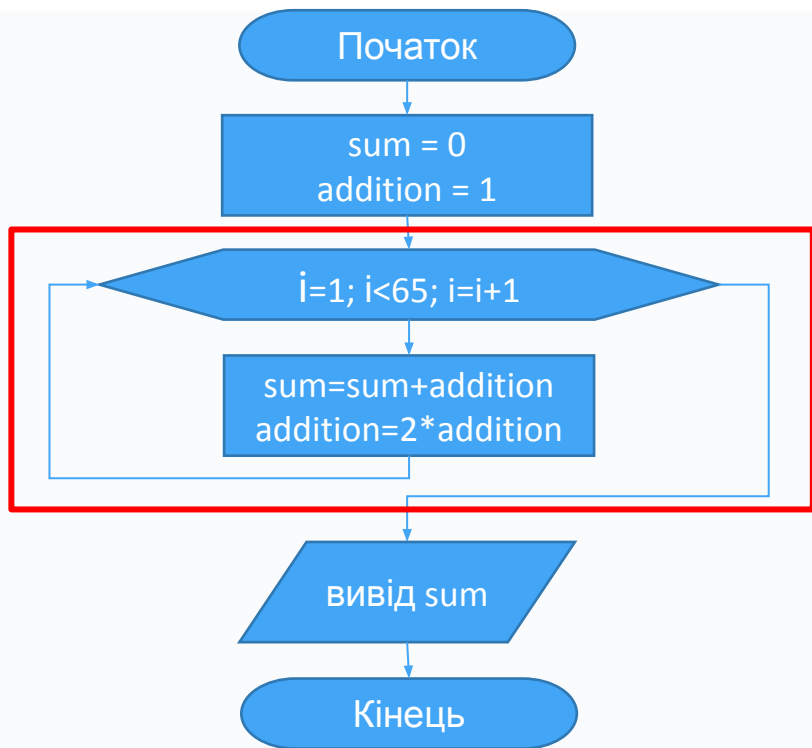
Практика



```
loops1.py X
loops1.py > ...
1  sum = 0
2  addition = 1
3
```



Практика



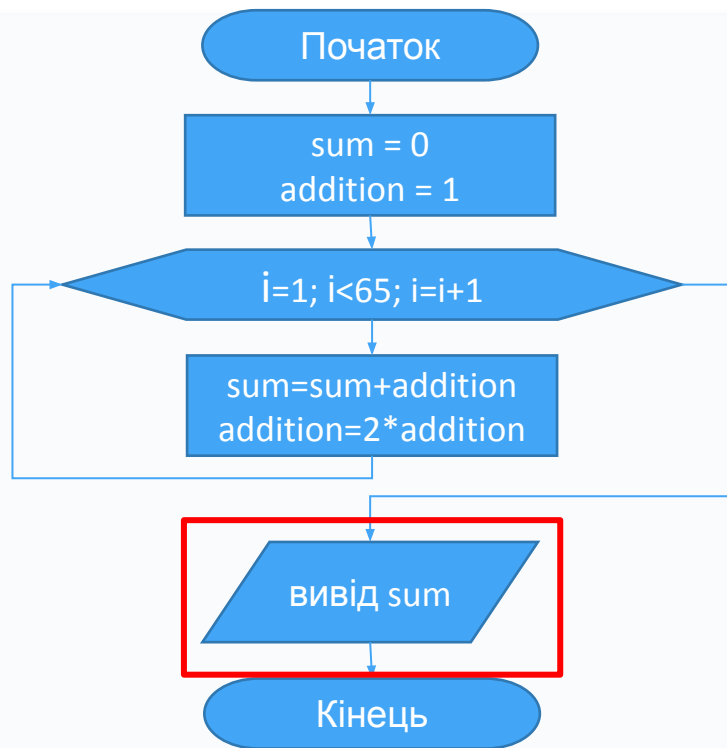
loops1.py ×

loops1.py > ...

```
1 sum = 0
2 addition = 1
3
4 for i in range(1, 65):
5     sum = sum + addition
6     addition = addition * 2
7
```



Практика



loops1.py ×

loops1.py > ...

```
1  sum = 0
2  addition = 1
3
4  for i in range(1, 65):
5      sum = sum + addition
6      addition = addition * 2
7
8  print("Кількість зерна =", sum)
9  |
```




Практыка



```
→ python_basics git:(master) x /usr/bin/python3  
y"  
Кількість зерна = 18446744073709551615
```



Підсумки



Цикли поділяють на два основні види:

цикл з параметром

цикл “поки”



Тіло циклу - дії, які повторюються щоразу при виконання циклу.

Параметр циклу - величина, від значення якої залежить, чи буде виконуватися тіло циклу.



Цикли повинні підкорятися трьом принципам:

- Циклічний процес повторює (ітерує) певний підпроцес – тіло циклу.
- Кожна ітерація повинна змінювати принаймні одне значення (параметр циклу)
- Повинна бути якась умова, за якої цикл завершується. Виконуючи цикл, система повинна досягати цього стану.