



Become QA Auto

Вказівка розгалуження (задача)

Бутенко Сергій

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

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

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

```
x = 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"
```

```
s = "Hello World"
```

```
x = 453
```

```
int(x)
```

```
int(y)
```

```
s = "Life Up"
```





План лекції



**Гнучка модель розробки проєктів
Agile**



Протокол роботи алгоритму



**Постановка задачі, побудова
моделі**



Програмна реалізація



Блок-схема алгоритму



**Тестування програми на
правильність роботи**

○ Гнучка модель розробки проєктів Agile



Ключовий принцип - розробка через короткі ітерації (цикли), в кінці кожного з яких замовник отримує робочий код або продукт

Кожна ітерація збільшує функціонал програми.



Задача обчислення вартості заряджання електромобіля

1. В програму вводять кількість електроенергії, яку бажає придбати клієнт.
2. Відома вартість кіловату електроенергії.
3. Програма обчислює суму покупки.
4. Вводять суму, яку дає клієнт.
5. Програма визначає решту, яку потрібно повернути клієнту.



Постановка задачі



Недолік попередньої версії

Якщо покупець вносить менше коштів, ніж вартість покупки, то програма обчислює решту, яка є від'ємним числом.

Приклад:

вартість покупки 500 грн.
внесені кошти 300 грн.
решта $300 - 500 = -200$ грн.

Завдання

Модифікувати розроблений алгоритм, щоб програма перевіряла, чи внесених коштів достатньо для оплати покупки, якщо так - обчислити решту.

Вхідні дані

- кількість електроенергії, яку купує клієнт,
- вартість одного кіловата електроенергії
- сума, надана клієнтом для оплати.

Шуканий результат

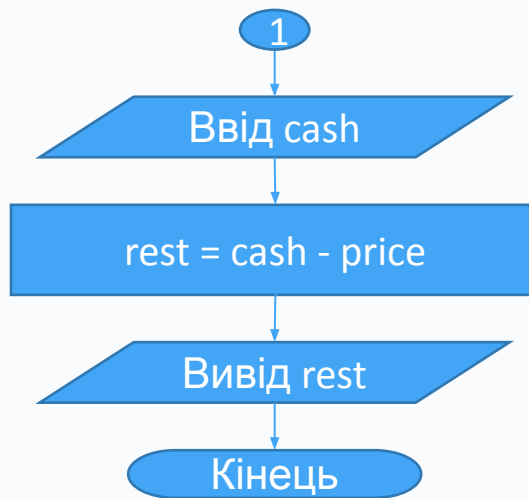
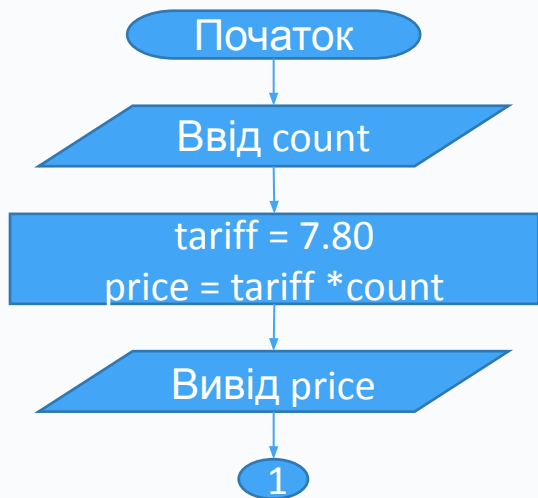
ціна покупки та решта, яку потрібно повернути клієнту.

Зв'язок між аргументами і результатом

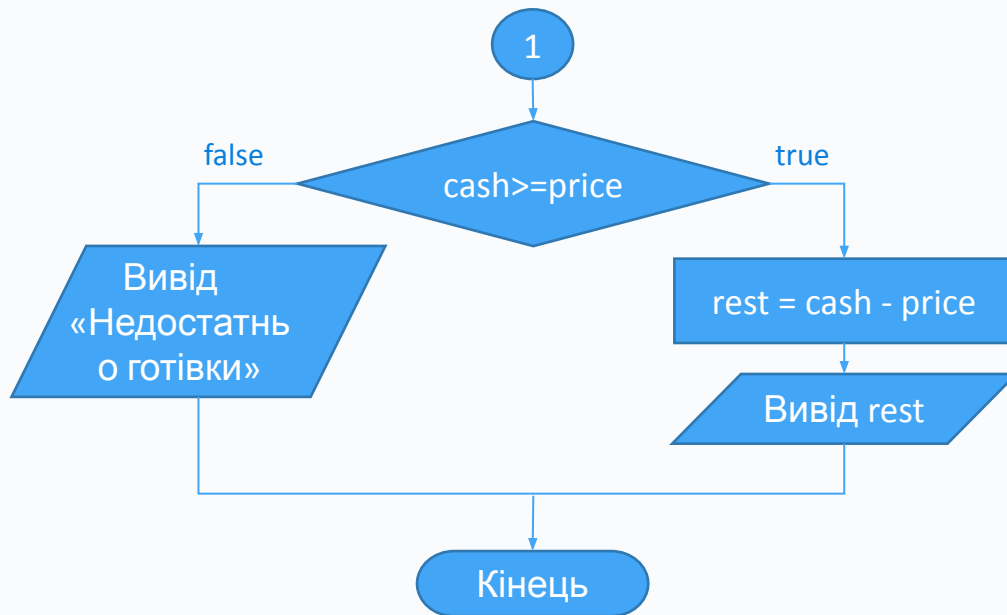
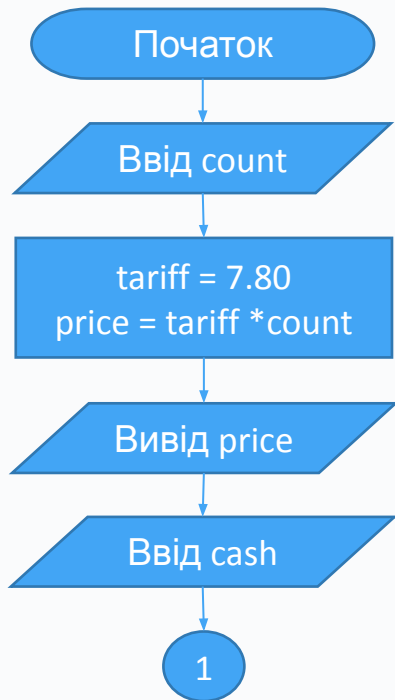
```
price = tariff * count
```

```
Якщо cash >= price, то rest = cash - price
```

⦿ Блок-схема алгоритму попередньої ітерації

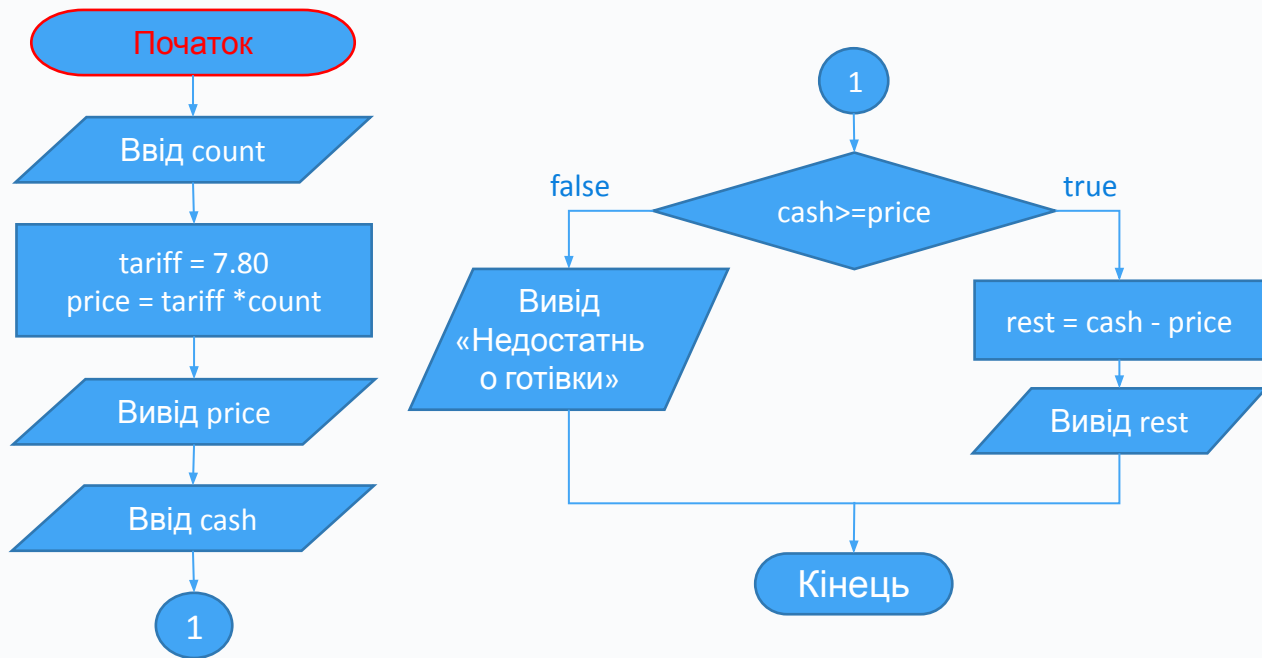


Блок-схема модифікованого алгоритму





Протокол виконання алгоритму

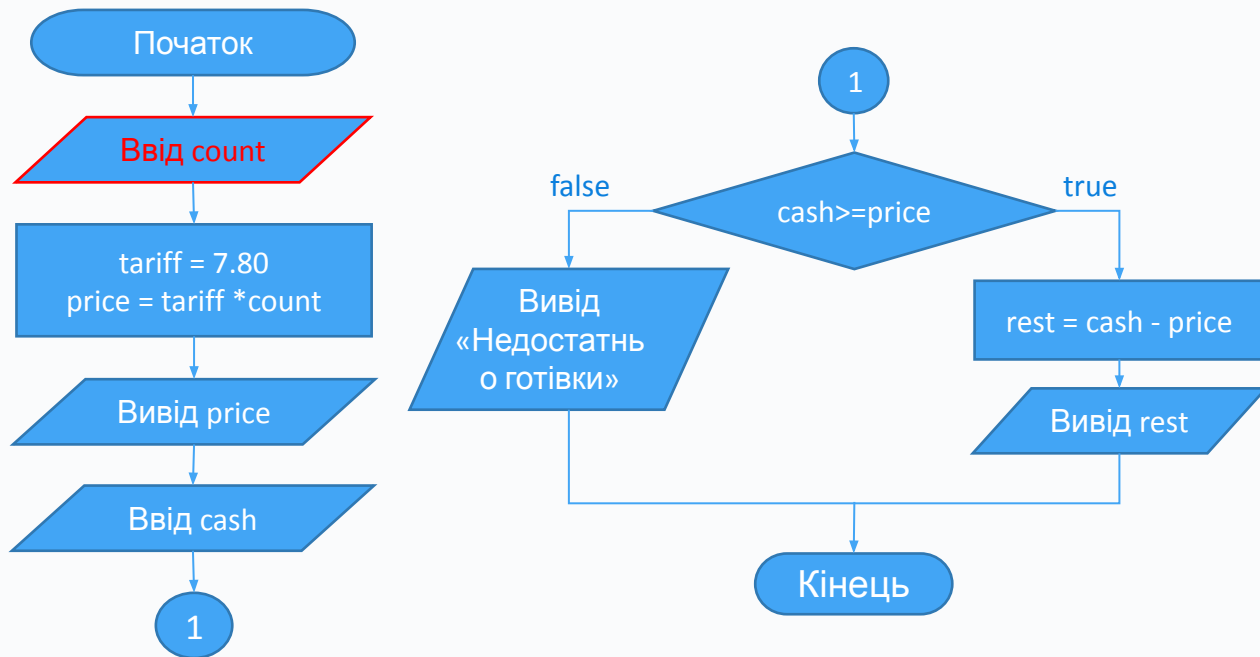


Вміст пам'яті та
результат логічних виразів

count	
tariff	
price	
cash	
rest	
cash >= price	



Протокол виконання алгоритму

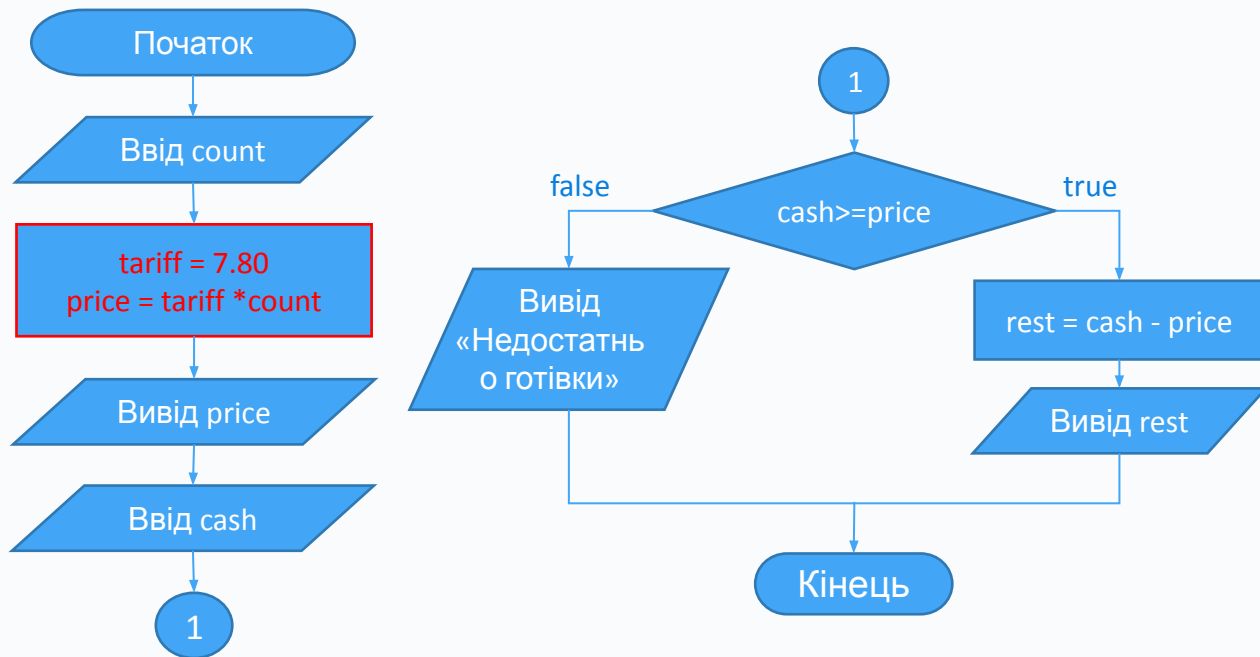


Вміст пам'яті та
результат логічних виразів

count	60
tariff	
price	
cash	
rest	
cash >= price	



Протокол виконання алгоритму

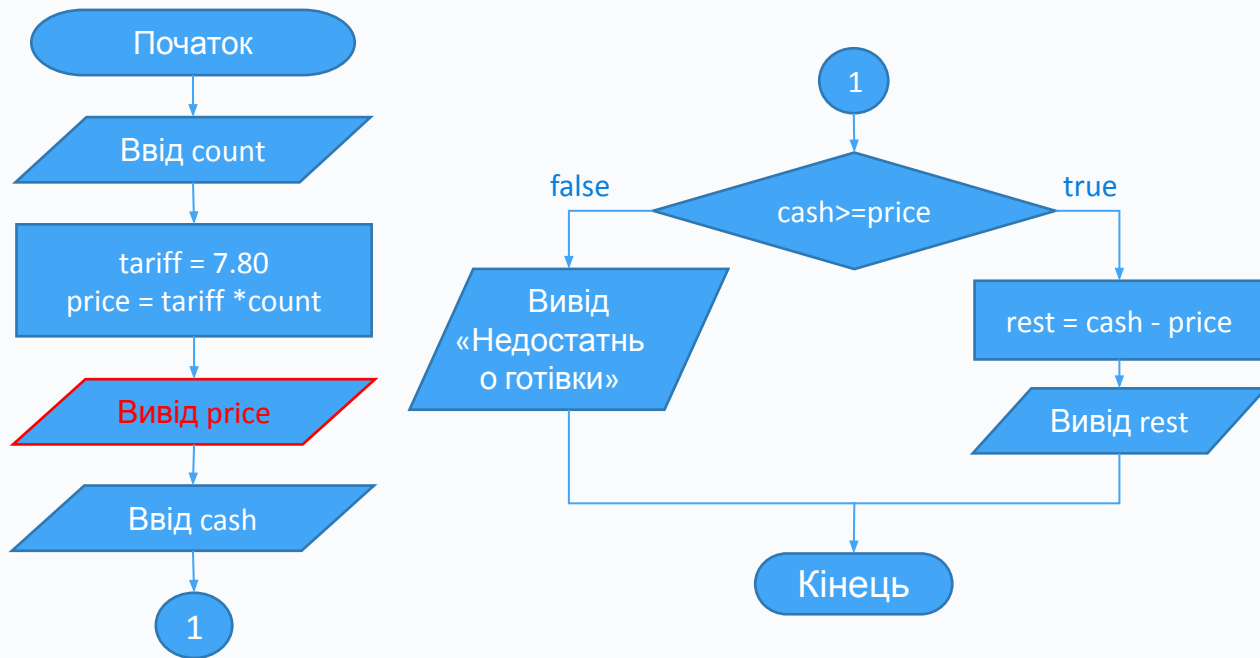


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	
rest	
cash >= price	



Протокол виконання алгоритму

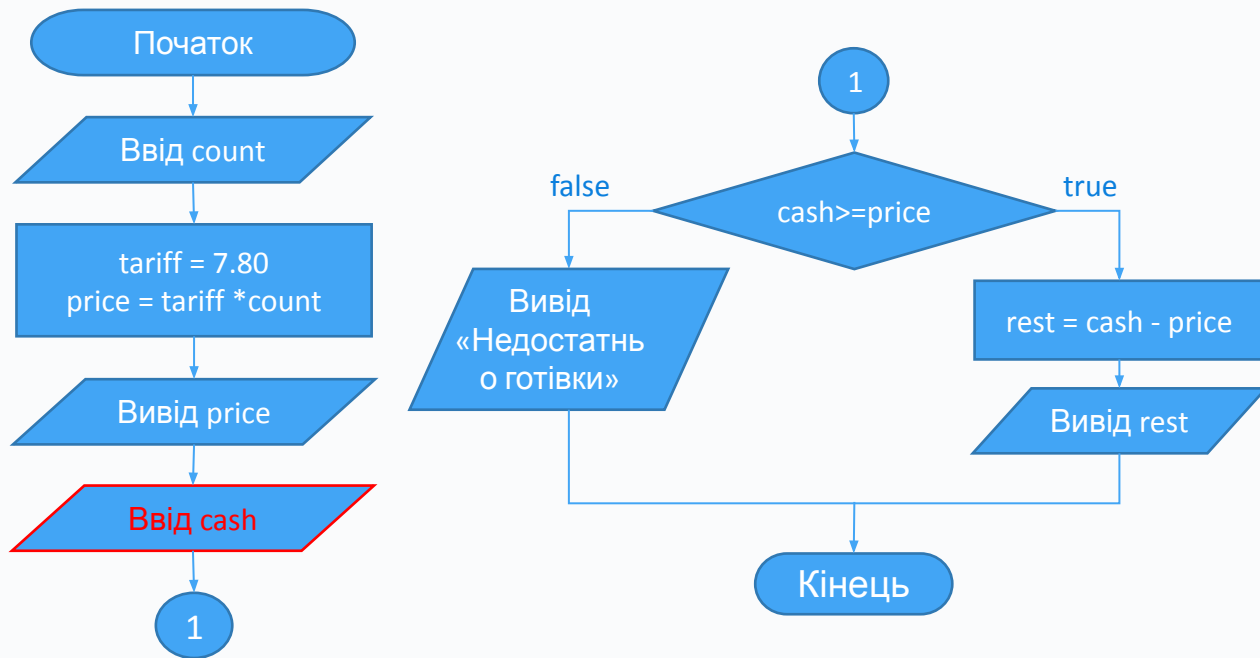


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	
rest	
$\text{cash} \geq \text{price}$	



Протокол виконання алгоритму

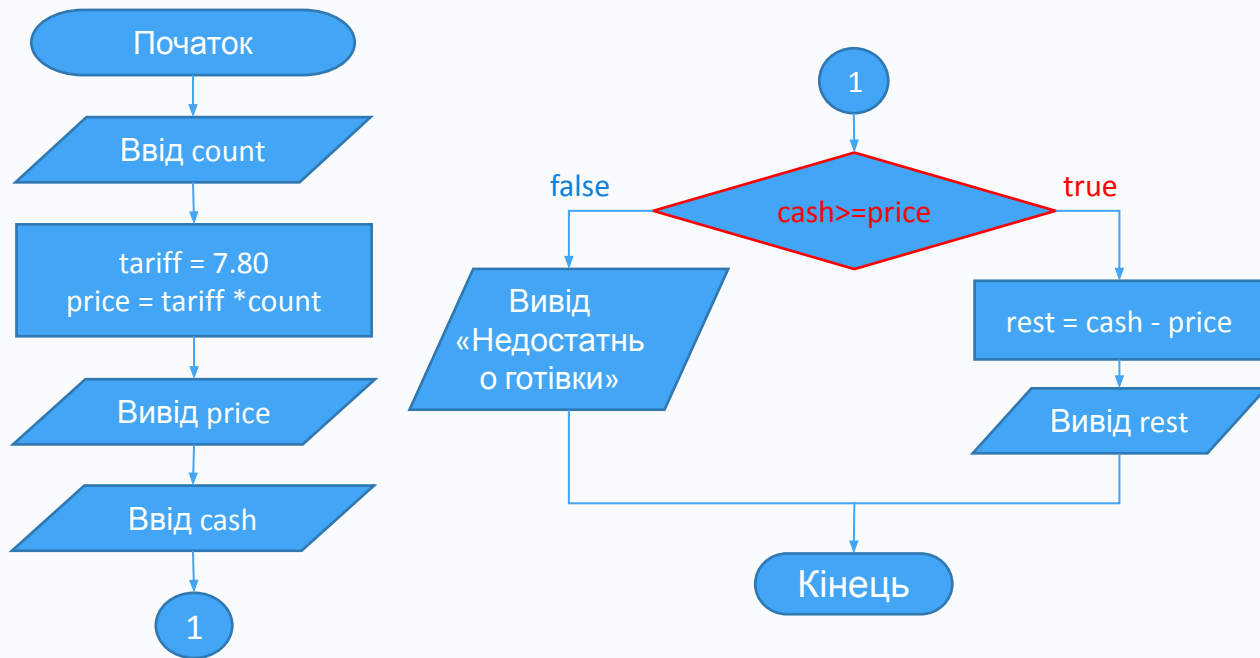


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	600
rest	
cash ≥ price	



Протокол виконання алгоритму

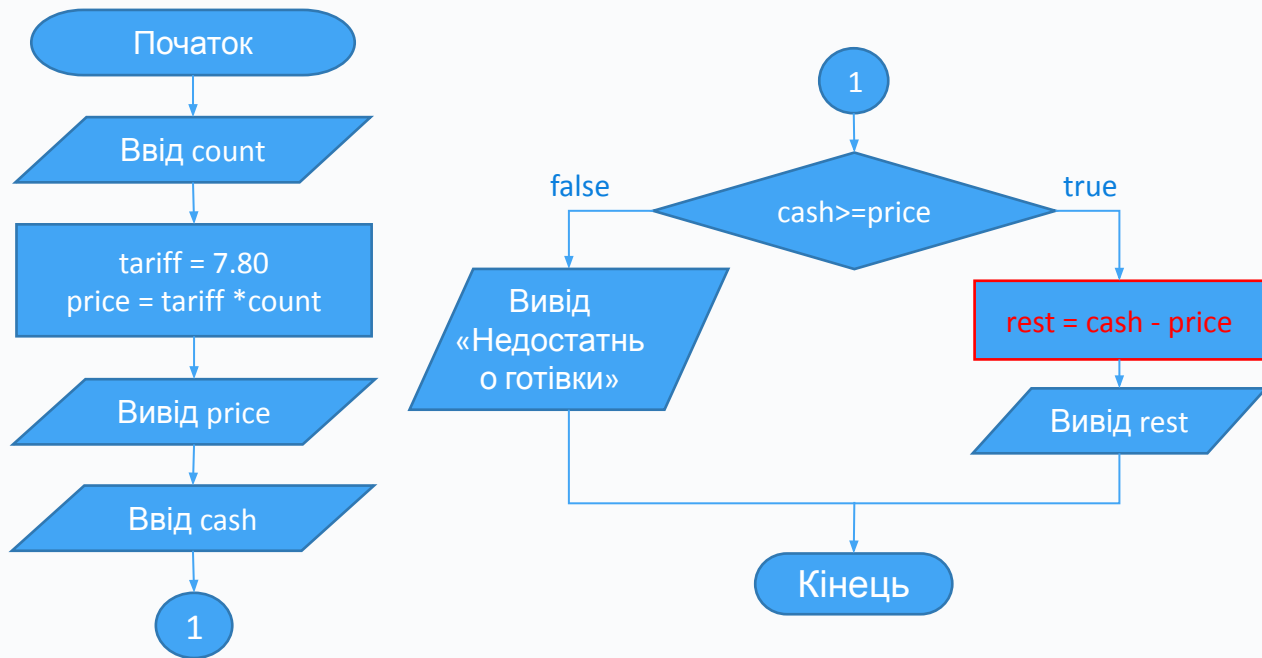


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	600
rest	
$\text{cash} \geq \text{price}$	true



Протокол виконання алгоритму

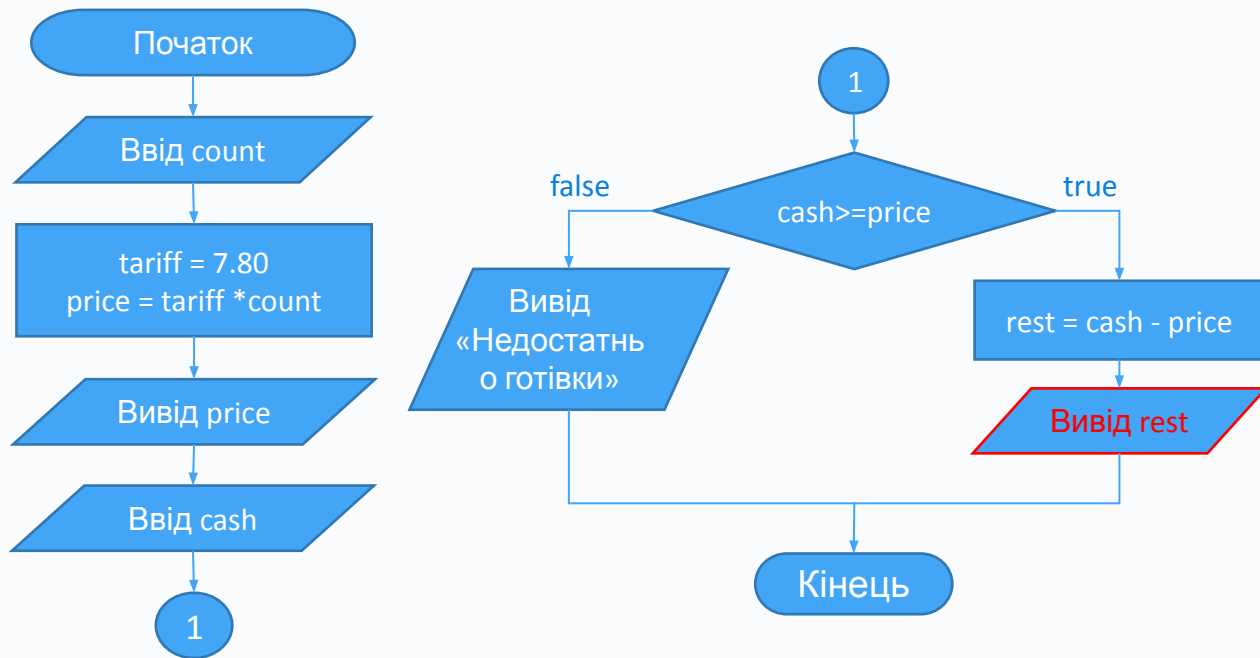


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	600
rest	132
cash >= price	true



Протокол виконання алгоритму

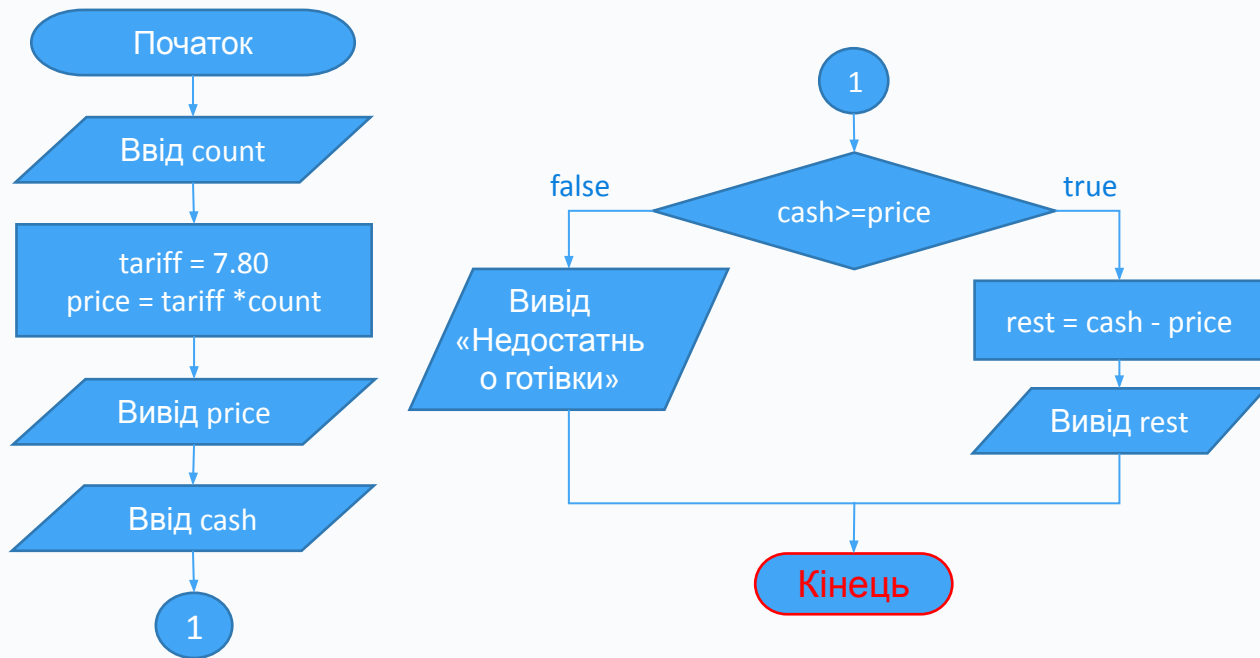


Вміст пам'яті та
результат логічних виразів

count	60
tariff	7.80
price	468
cash	600
rest	132
$\text{cash} \geq \text{price}$	true



Протокол виконання алгоритму

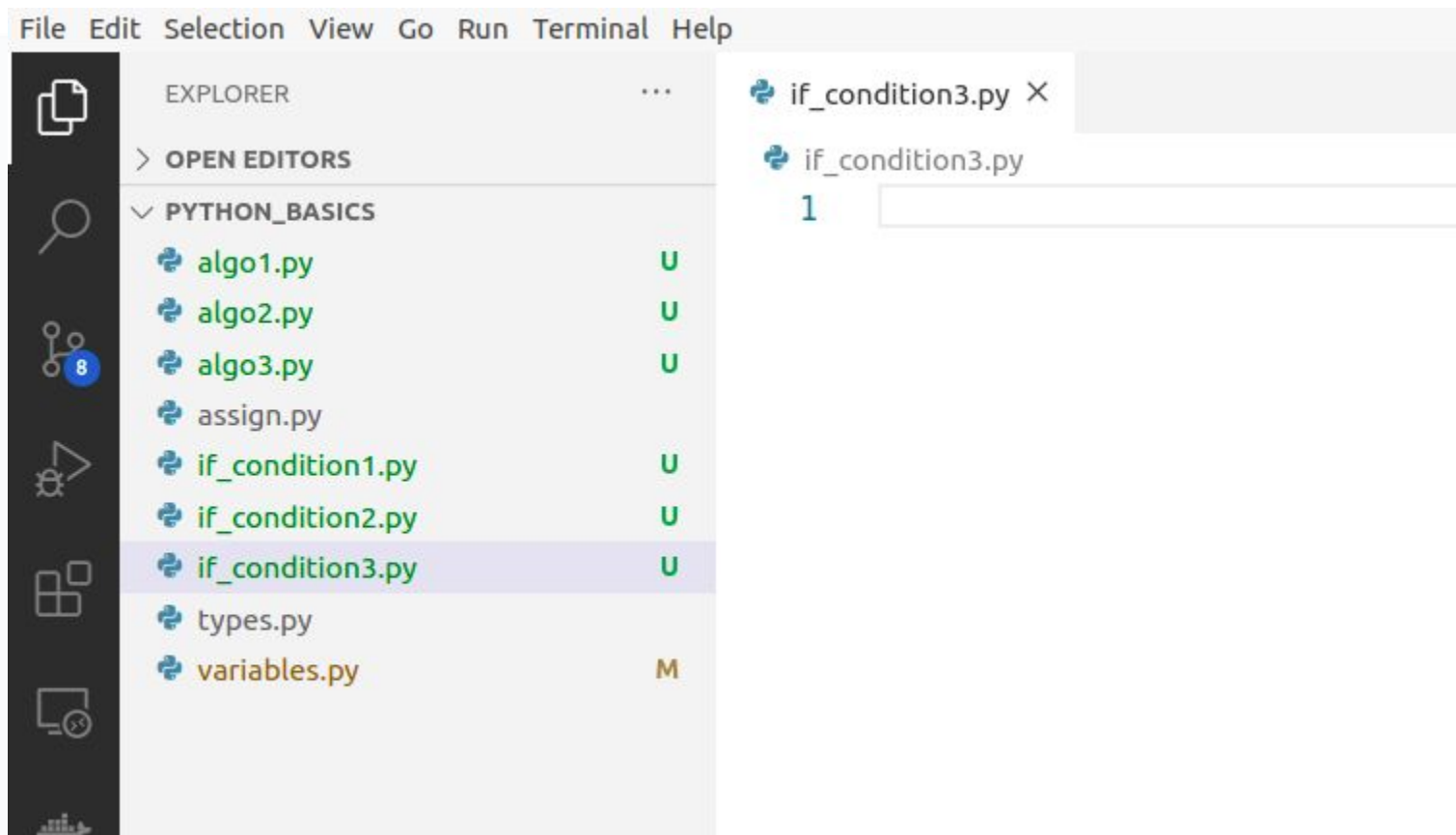


Вміст пам'яті та
результат логічних виразів

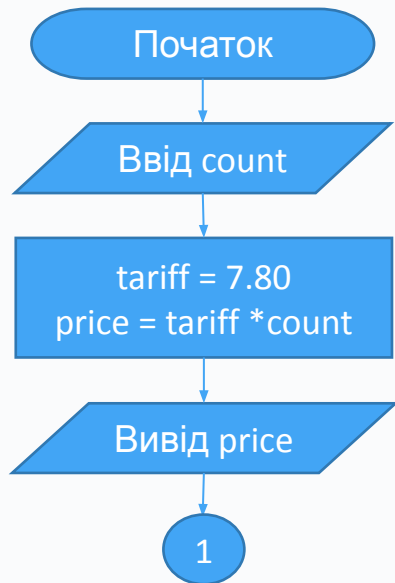
count	60
tariff	7.80
price	468
cash	600
rest	132
cash >= price	true



Практика

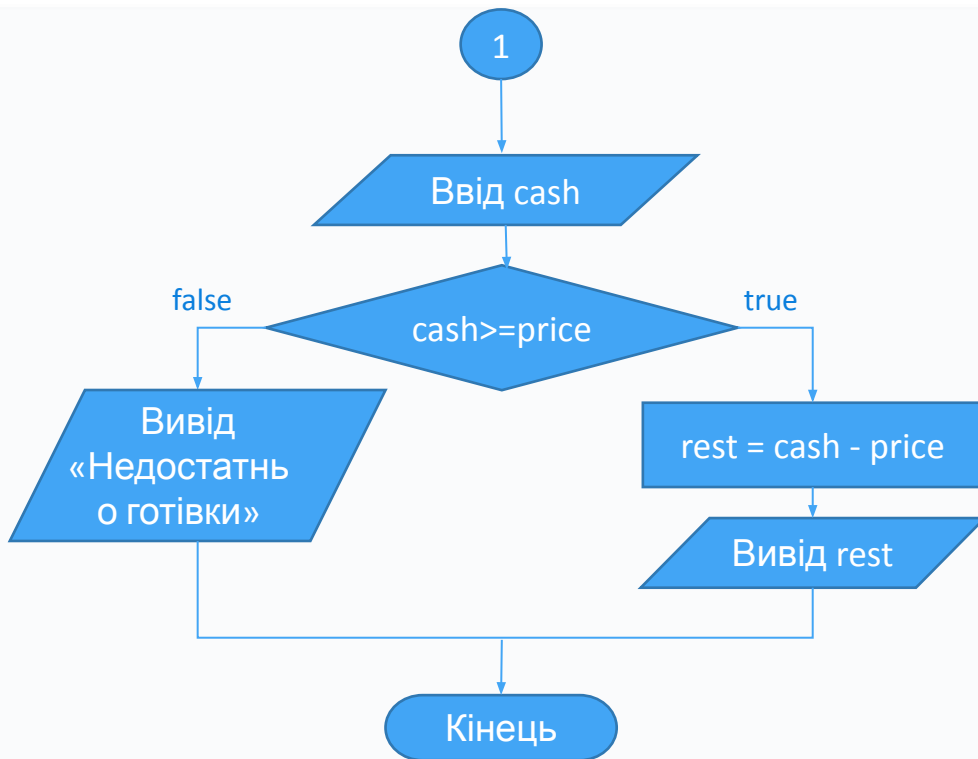


Практика



```
1 count = float(input("Введіть бажану кількість електроенергії: "))
2 tariff = 7.80
3 price = tariff * count
4 print("Ціна заряджання =", round(price, 2))
```

Блок-схема модифікованого алгоритму



```
1 count = float(input("Введіть бажану кількість електроенергії: "))
2 tariff = 7.80
3 price = tariff * count
4 print("Ціна заряджання =", round(price, 2))
5
6 cash = float(input("Введіть суму готівки: "))
7 if cash >= price:
8     rest = cash - price
9     print("Решта =", round(rest, 2))
10 else:
11     print("Недостатньо готівки")
12
```



Тестування програми



Можливі комбінації вхідних даних:

Введіть бажану кількість електроенергії
60
Ціна заряджання = 468.0
Введіть суму готівки
600
Решта = 132.0

Внесена сума коштів є більшою за ціну замовлення

count = 60
cash = 600

Введіть бажану кількість електроенергії
70
Ціна заряджання = 546.0
Введіть суму готівки
400
Недостатньо готівки

Внесена сума коштів є меншою за ціну замовлення

count = 70
cash = 400

Введіть бажану кількість електроенергії
50
Ціна заряджання = 390.0
Введіть суму готівки
390
Решта = 0.0

Внесена сума коштів дорівнює ціні замовлення

count = 50
cash = 390



Підсумки



Застосували гнучку модель розробки проєктів Agile і виконали другу ітерацію для задачі обчислення вартості заряджання електромобіля.



Розширили функціонал: програма перевіряє, чи внесених коштів достатньо для оплати покупки, якщо так - обчислює решту.



Дослідили роботу алгоритму за допомогою протоколу його виконання.



Виконали програмну реалізацію алгоритму і протестували програму на правильність результатів.