

Урок 19. Дані різних типів

Вивчення нового матеріалу

Слайд № 1

Ми вже знаємо, що відповіді користувача можна зберігати у змінних – своєрідних контейнерах. Але виявляється, що залежно від того, який це контейнер, ми можемо зберігати в ньому різні дані й виконувати над ними різні операції.



Запам'ятай: над різними даними виконуються різні операції.

Слайд № 2

Розглянемо невелику програму, що додає два числа, які користувач вводить з клавіатури.

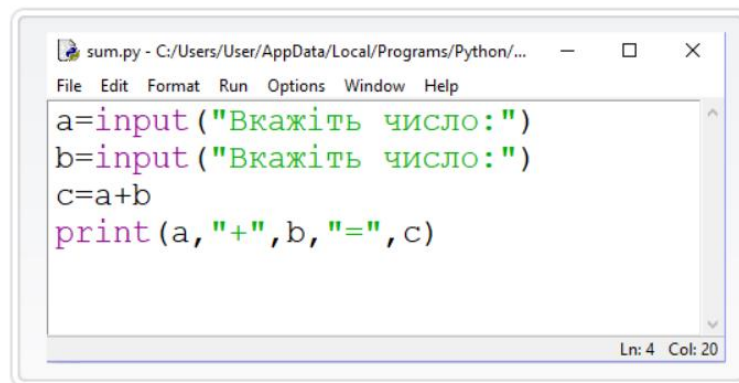
```
sum.py - C:/Users/User/AppData/Local/Programs/Python/...  
File Edit Format Run Options Window Help  
a=input("Вкажіть число:")  
b=input("Вкажіть число:")  
c=a+b  
print(a, "+", b, "=", c)  
Ln: 4 Col: 20
```



Олесь покаже, як створити та виконати цю програму в середовищі Python.

Слайд № 3

Розглянемо невелику програму, що додає два числа, які користувач вводить з клавіатури.



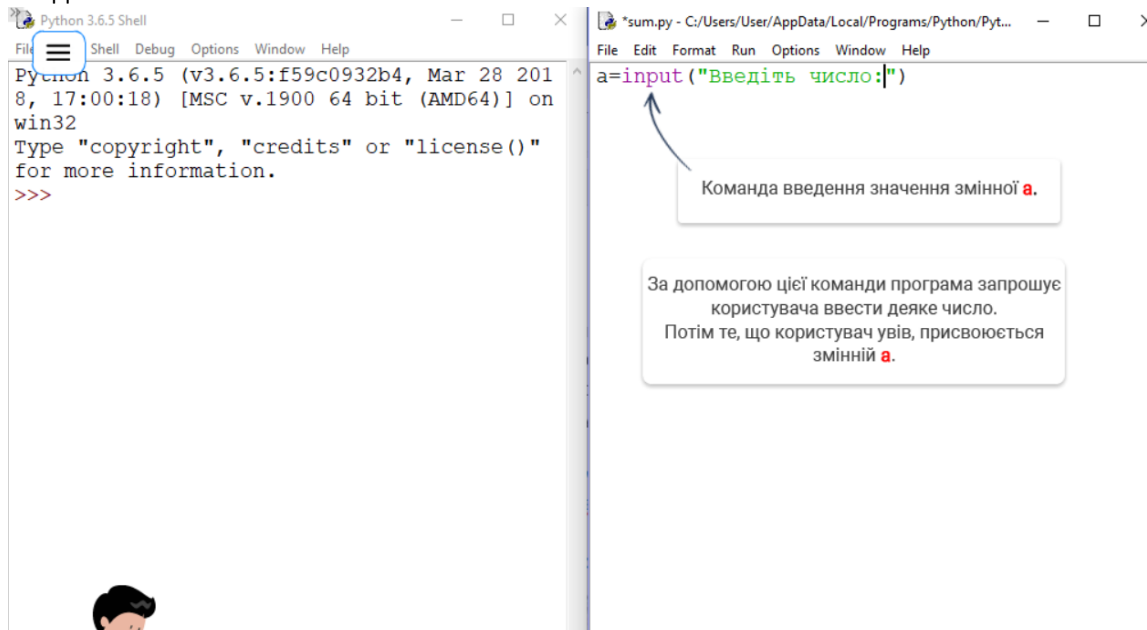
```
sum.py - C:/Users/User/AppData/Local/Programs/Python/...
File Edit Format Run Options Window Help
a=input("Вкажіть число:")
b=input("Вкажіть число:")
c=a+b
print(a, "+", b, "=", c)
Ln: 4 Col: 20
```



Одесь покаже, як створити та виконати цю програму в середовищі Python.

Слайд № 4

Введення змінної **a**.



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

*sum.py - C:/Users/User/AppData/Local/Programs/Python/Pyt...
File Edit Format Run Options Window Help
a=input("Введіть число:|")

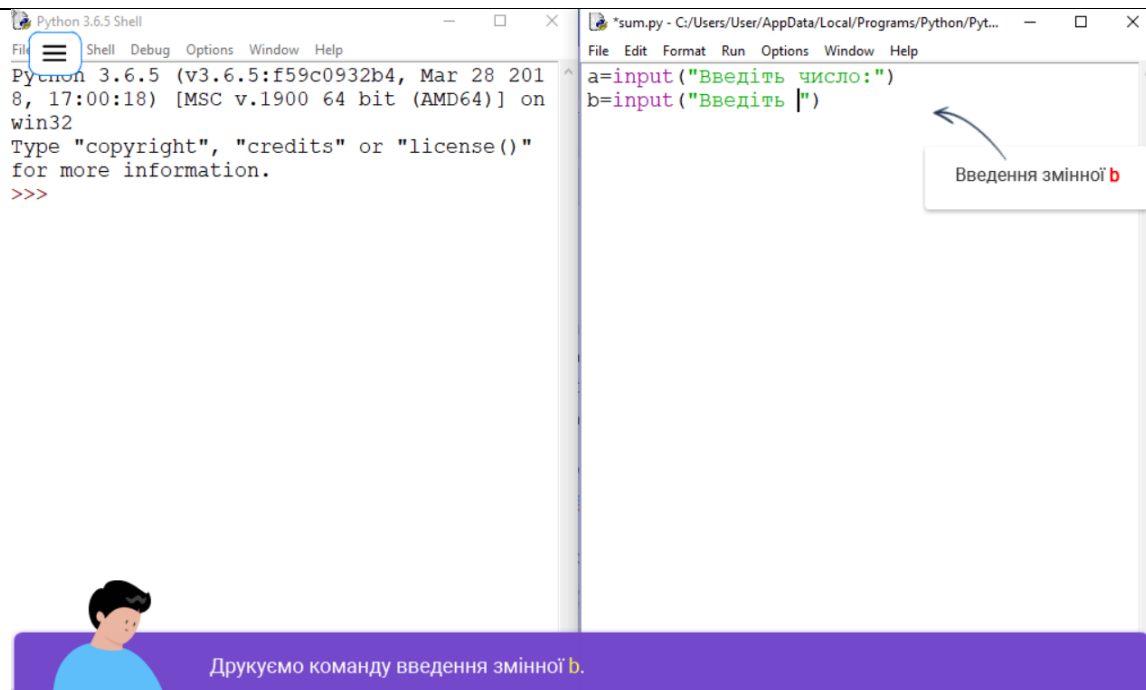
Команда введення значення змінної **a**.

За допомогою цієї команди програма запрошує користувача ввести деяке число. Потім те, що користувач увів, присвоюється змінній **a**.



Відповідь користувача — це і буде значення змінної **a**.

Слайд № 5



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

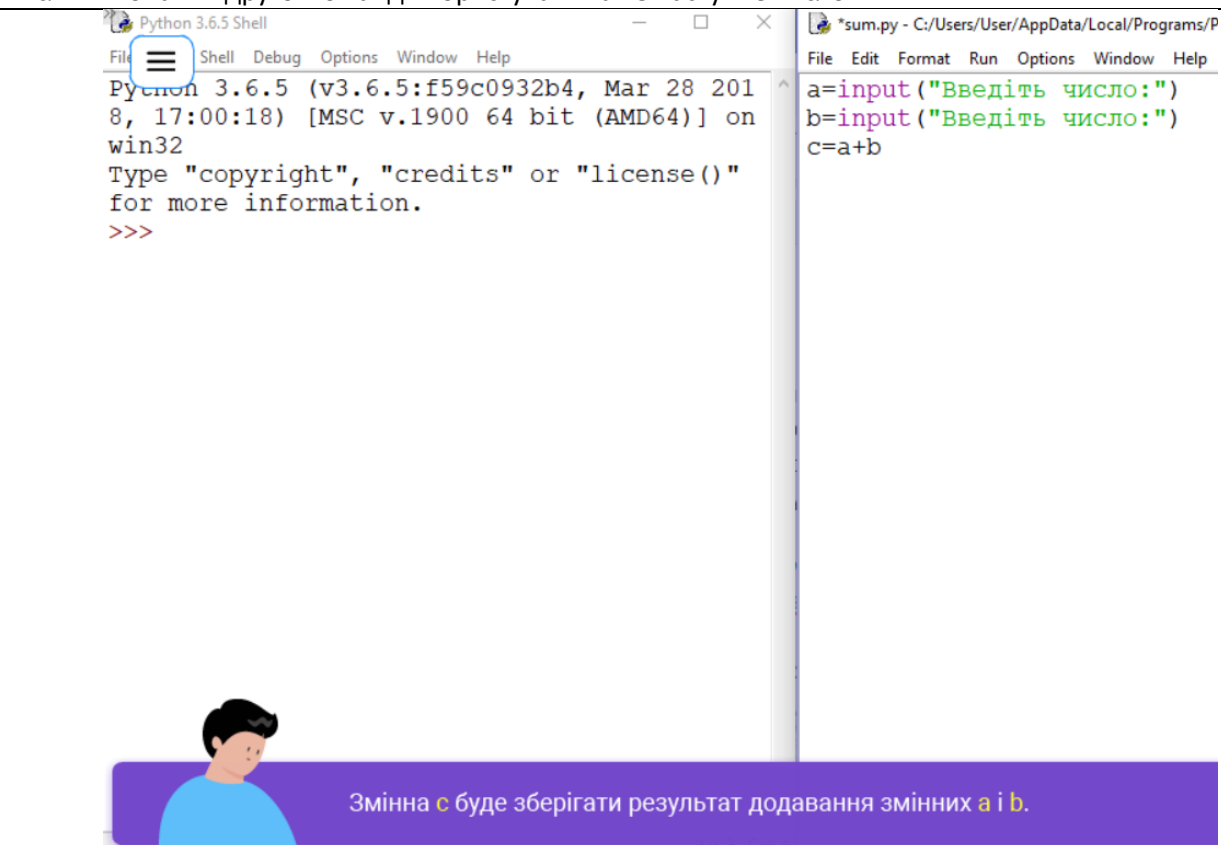
File Edit Format Run Options Window Help
a=input("Введіть число:")
b=input("Введіть ")

Введення змінної **b**

Друкуємо команду введення змінної **b**.

Після виконання другої команди користувач вкаже наступне число.

Слайд № 6

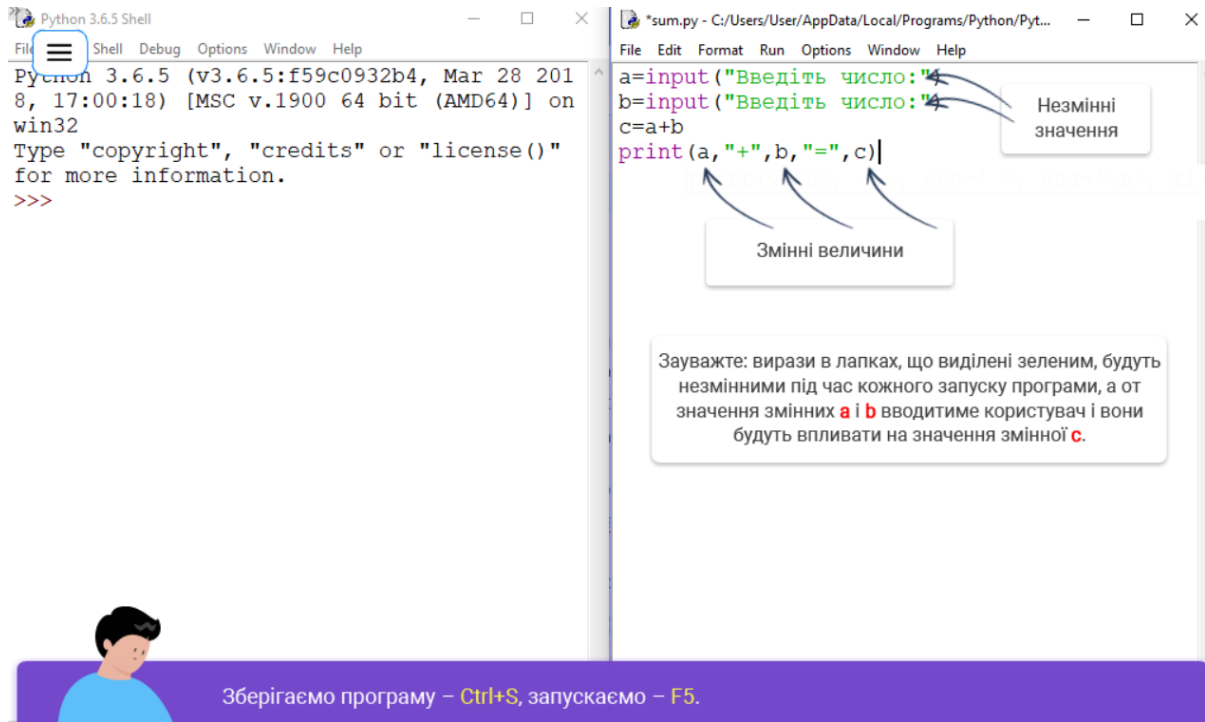


Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

File Edit Format Run Options Window Help
a=input("Введіть число:")
b=input("Введіть число:")
c=a+b

Змінна **c** буде зберігати результат додавання змінних **a** і **b**.

Слайд № 7



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

File Edit Format Run Options Window Help

```
a=input("Введіть число:")  
b=input("Введіть число:")  
c=a+b  
print(a,"+",b,"=",c)
```

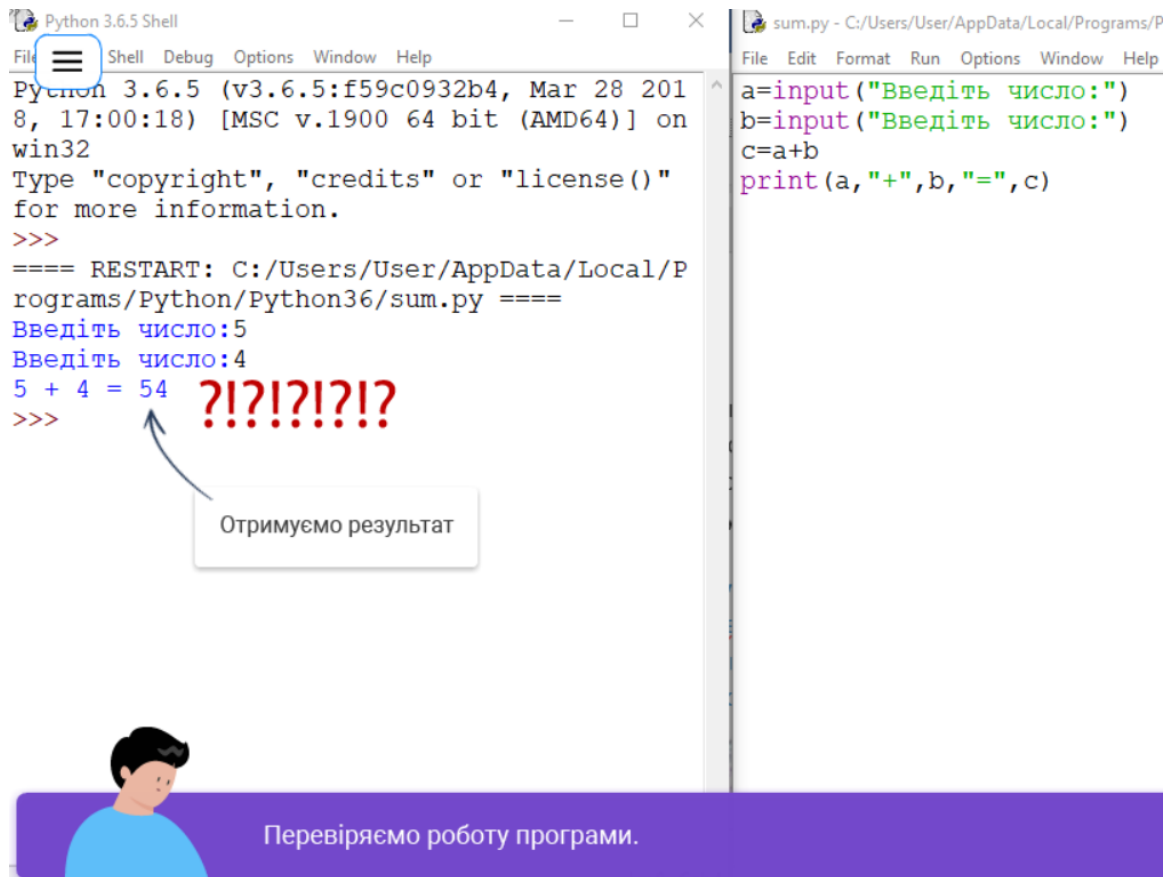
Незмінні значення

Змінні величини

Зауважте: вирази в лапках, що виділені зеленим, будуть незмінними під час кожного запуску програми, а от значення змінних **a** і **b** вводитиме користувач і вони будуть впливати на значення змінної **c**.

Зберігаємо програму – **Ctrl+S**, запускаємо – **F5**.

Слайд № 8



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python36/sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 54
>>>

File Edit Format Run Options Window Help

```
a=input("Введіть число:")  
b=input("Введіть число:")  
c=a+b  
print(a,"+",b,"=",c)
```

Отримуємо результат

Перевіряємо роботу програми.

Звісно, Python вміє додавати! Давайте розберемося, чому в Python може вийти "5"+"4"=54.

Ми вказали, що будемо вводити щось з клавіатури, і команда **input** просто приймає дані. Зберегти ці дані команда може у змінній.

```
a=input("Введіть число:")
```



Змінні не тільки називаються по-різному, але й розрізняються за своїми можливостями.

Змінні – це приблизно, як коробки різної форми: вони можуть зберігати різні речі. Залежно від того, якої форми ця коробка, з нею можна виконувати різні дії.



Запам'ятай: зі змінними різних типів можна виконувати різні дії.

Наприклад, коробки з піцею можна легко скласти одна на одну.
А от коробочки-пірамідки – ні.



Запам'ятай: зі змінними різних типів можна виконувати різні дії.

На сьогодні нам достатньо знати, що бувають змінні для зберігання тексту та чисел.
Причому числа бувають цілі та дробові.



Запам'ятай: у змінних можна зберігати текст та числа.

Слайд № 13

Ввівши значення з клавіатури у відповідь на команду **input**, ми просто отримаємо певний текст. І навіть якщо він містить числа, це будуть просто символи. Зберігає їх змінна типу **string**.



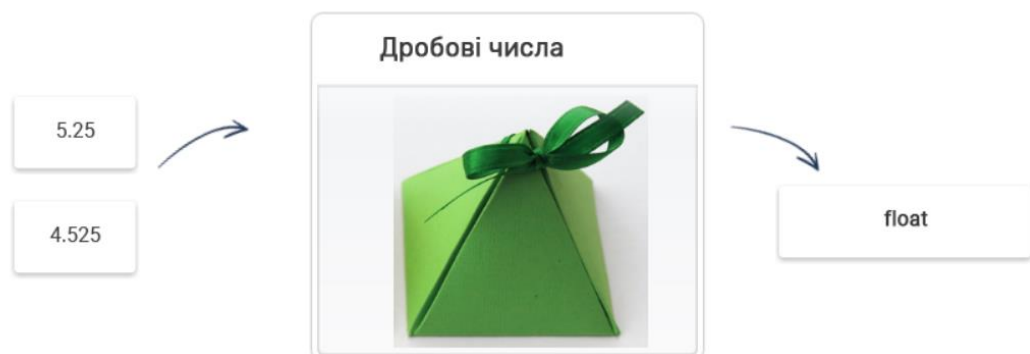
Запам'ятай: введені з клавіатури значення мають тип **string**.

Слайд № 14



Змінні, що містять цілі числа, мають тип **int** — від англійського слова **integer** (ціле число).

Слайд № 15



Змінні, що містять дробові числа, мають тип **float**.

За допомогою спеціальних команд можна текст перетворити на число.

`y=int(x)` – перетворює текстову змінну **x** на цілочисельну змінну **y**

`y=float(x)` – перетворює текстову змінну **x** на дробову змінну **y**

Наприклад

`a = int(input("Введи число"))` → `a` – ціле число

`a = float(input("Введи число"))` → `a` – дробове число



Запам'ятай спеціальні функції для перетворення тексту на число.

Коли ми ввели текст **4** та **5**, Python склав цей текст, приписуючи символи один до одного.

`"5"+"4"="54"`

Це навіть не помилка, ми ж не побачили ніяких червоних зауважень. Шкода, але такі помилки Python не допомагає нам відловлювати.

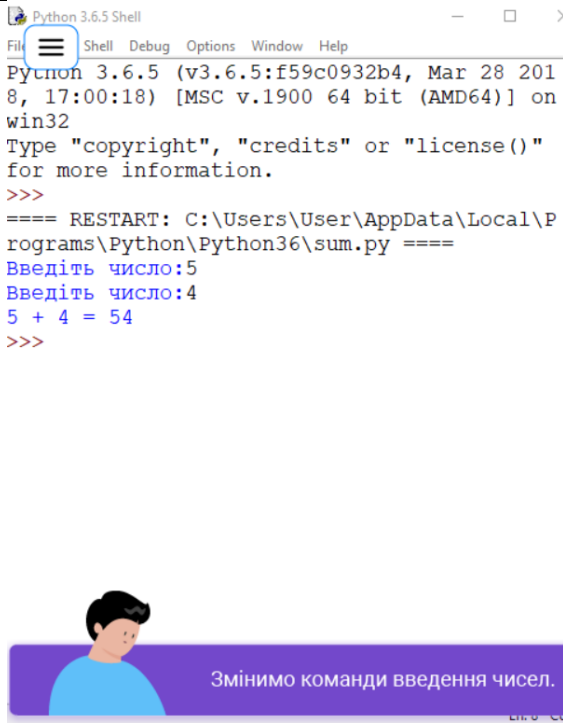
Отже, усі програми ми маємо перевіряти на відповідність поставленим вимогам самостійно.

Якщо програма мала додавати два числа, то її результатом має бути сума цих чисел, а не якась абракадабра.

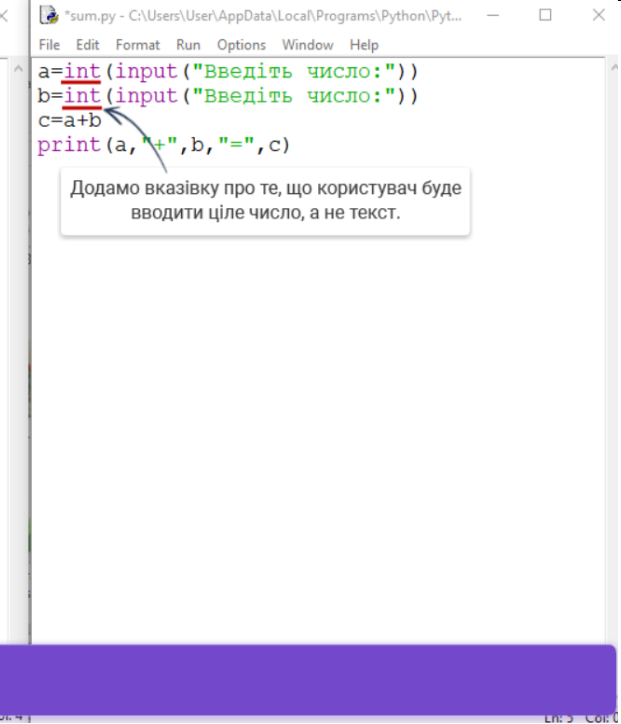


Далі Олесь продемонструє, як уникнути такої помилки.

Слайд № 18



```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 54
>>>
```



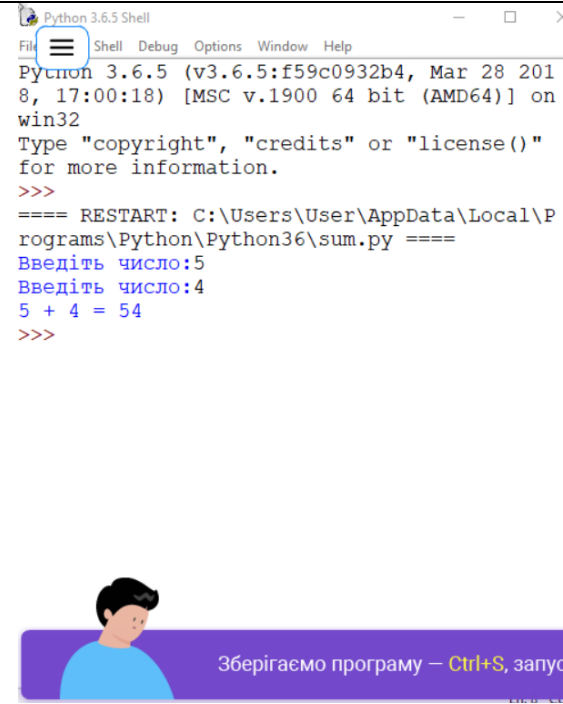
```
*sum.py - C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py
File Edit Format Run Options Window Help
a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```

Додамо вказівку про те, що користувач буде вводити ціле число, а не текст.



Змінимо команди введення чисел.

Слайд № 19



```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 54
>>>
```



```
*sum.py - C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py
File Edit Format Run Options Window Help
a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```



Зберігаємо програму — **Ctrl+S**, запускаємо — **F5**.

Python 3.6.5 Shell
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 54
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4

sum.py - C:\Users\User\AppData\Local\Programs\Python\Pyth...
a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)

Вводимо число 4

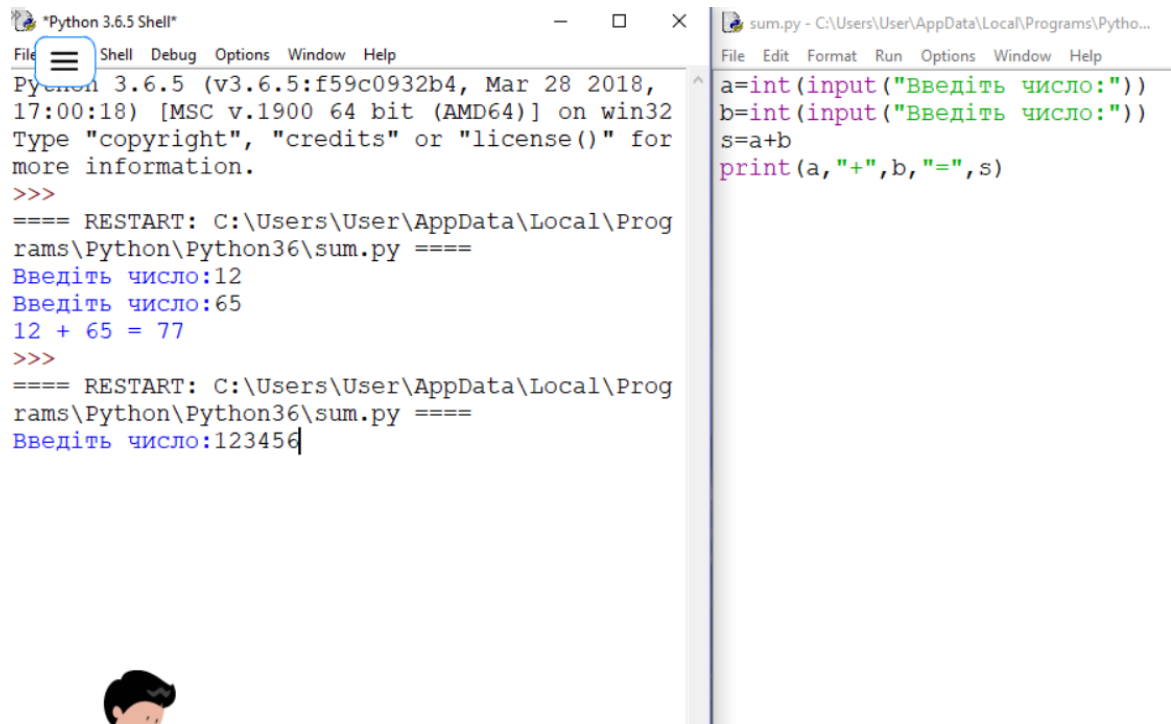
Перевіряємо роботу програми.

Python 3.6.5 Shell
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 54
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5
Введіть число:4
5 + 4 = 9
>>>

sum.py - C:\Users\User\AppData\Local\Programs\Python\Pyth...
a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)

На цей раз все правильно!

Зауважте, що програму можна виконувати багато разів, вводячи різні дані та отримуючи різні результати.



The screenshot shows two windows. The left window is the Python 3.6.5 Shell, and the right window is the sum.py file. The shell shows the execution of the program with two test cases: adding 12 and 65 to get 77, and adding 123456 and 654321 to get 777777. The sum.py file contains the following code:

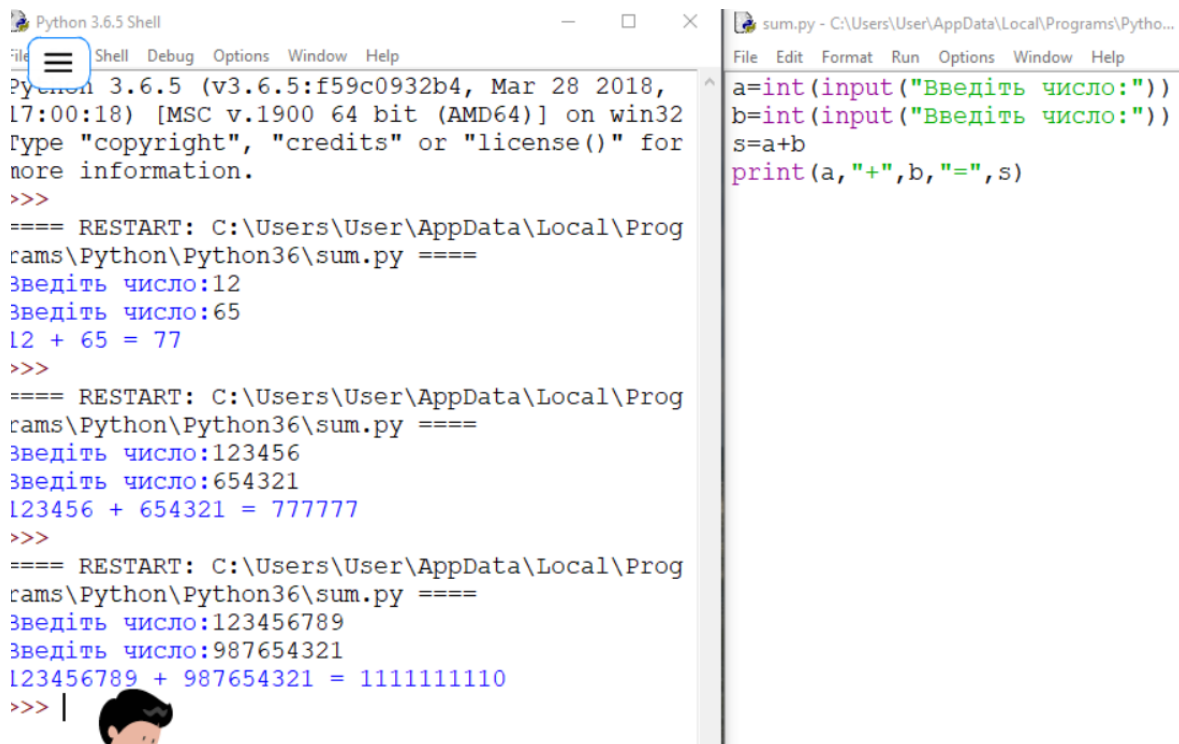
```

a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
s=a+b
print(a,"+",b,"=",s)

```



Python вміє виконувати розрахунки з достатньо великими числами.



The screenshot shows two windows. The left window is the Python 3.6.5 Shell, and the right window is the sum.py file. The shell shows the execution of the program with two test cases: adding 12 and 65 to get 77, and adding 123456 and 654321 to get 777777. The sum.py file contains the following code:

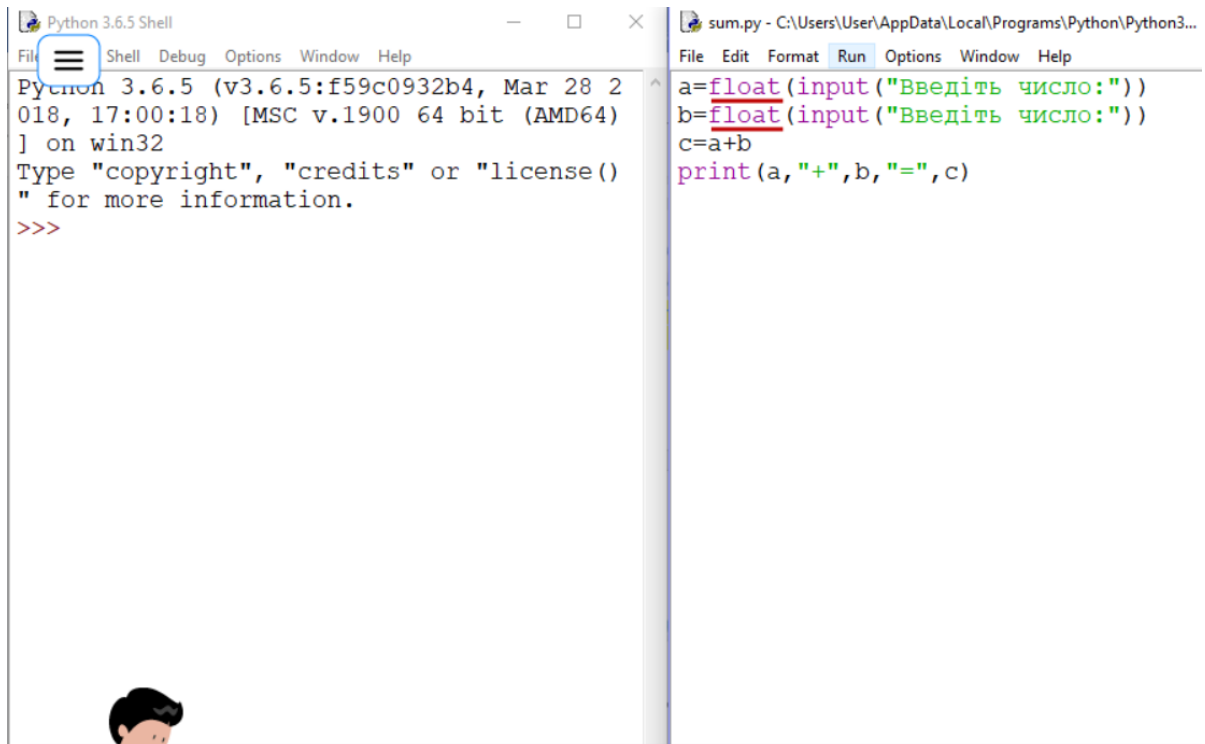
```

a=int(input("Введіть число:"))
b=int(input("Введіть число:"))
s=a+b
print(a,"+",b,"=",s)

```



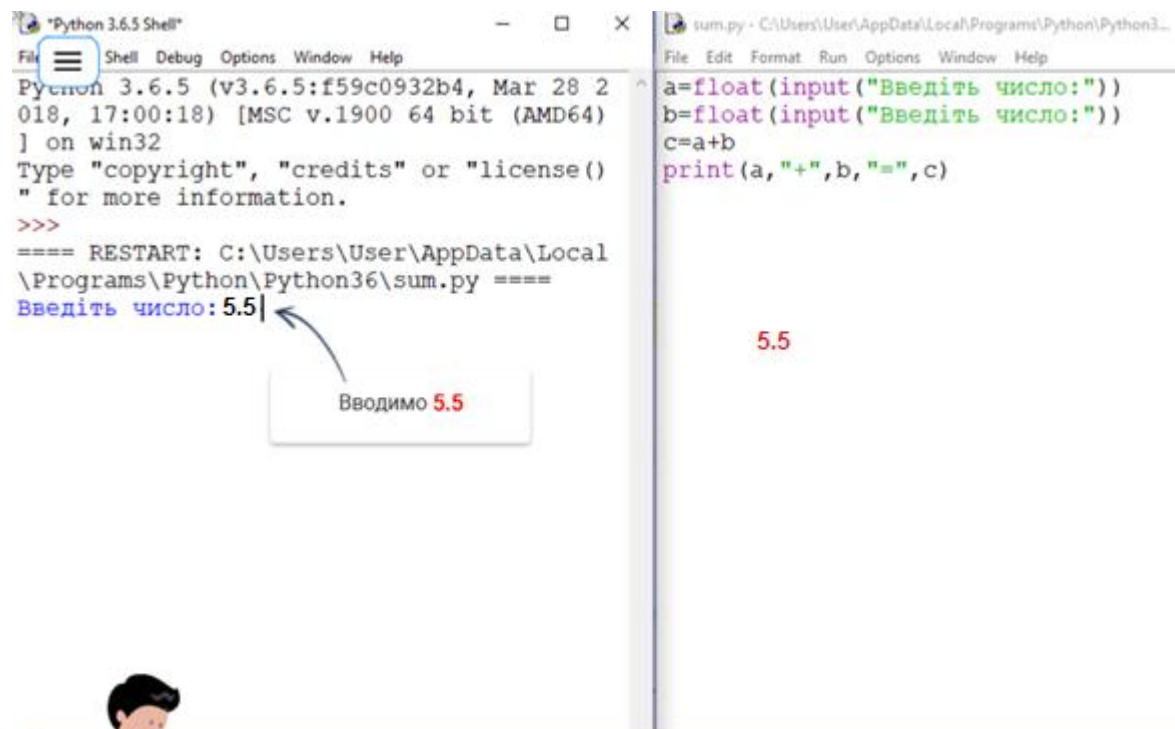
Python вміє виконувати розрахунки з достатньо великими числами, навіть більшими за ті, що є в калькуляторі.



The screenshot shows two windows. The left window is the Python 3.6.5 Shell, displaying the standard startup message. The right window is a code editor for a file named 'sum.py', containing the following Python code:

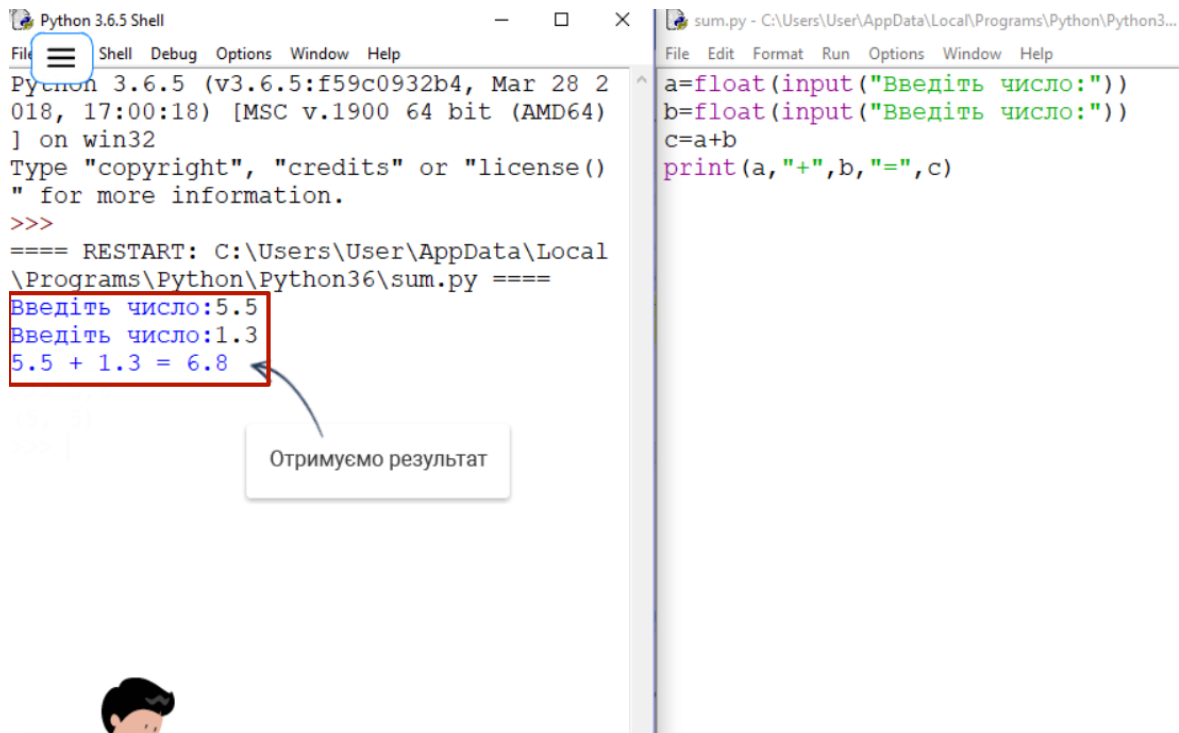
```
a=float(input("Введіть число:"))
b=float(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```

Below the code editor, a purple banner contains the text: Виконаємо дії з дробовими числами, записавши замість `int` команду `float`.



The screenshot shows the same two windows as in slide 24. In the Python 3.6.5 Shell, the user has entered the command `==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====` and is now at the prompt `Введіть число:5.5|`. An arrow points from a callout box containing the text "Вводимо 5.5" to the input. The code editor on the right shows the same code as before.

Below the code editor, a purple banner contains the text: Перевіряємо роботу програми.

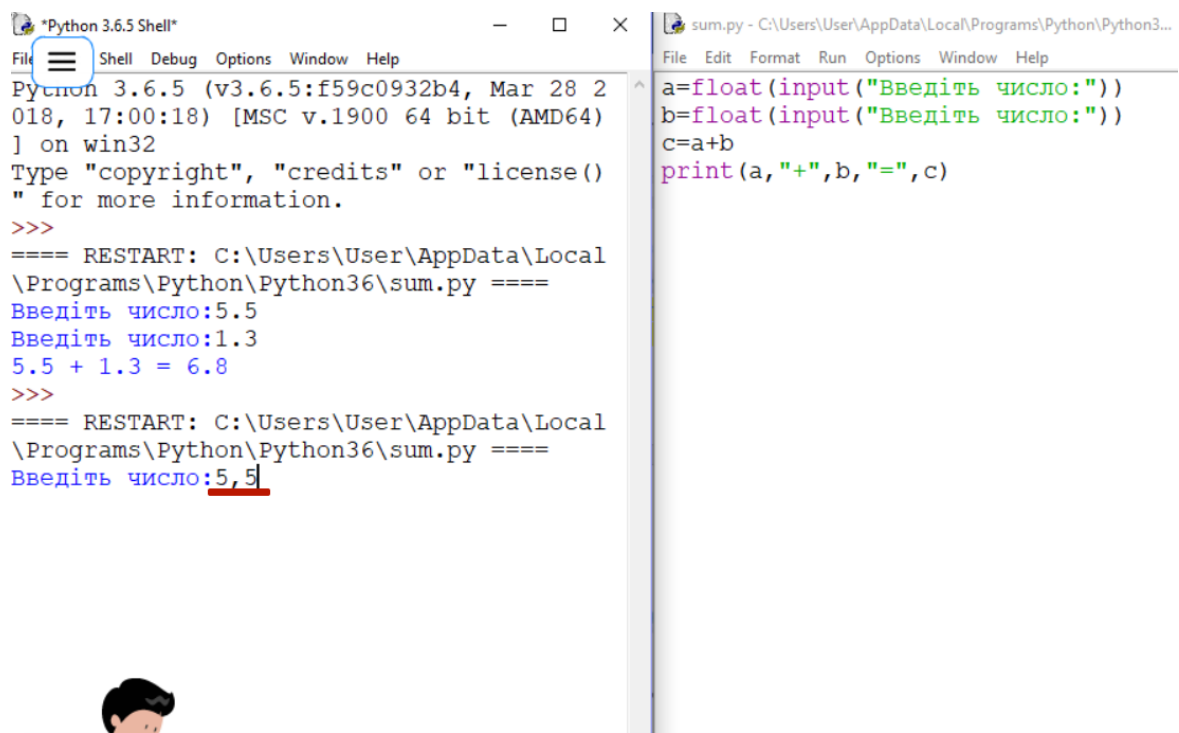


The screenshot shows two windows. The left window is the Python 3.6.5 Shell, and the right window is the sum.py file. The shell shows the execution of the script, which prompts for two numbers and prints their sum. The inputs are 5.5 and 1.3, and the output is 6.8. A red box highlights the input and output lines, and an arrow points to the result with the text 'Отримуємо результат' (We get the result).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5.5
Введіть число:1.3
5.5 + 1.3 = 6.8
```

```
a=float(input("Введіть число:"))
b=float(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```

Отримуємо результат.



The screenshot shows the same Python 3.6.5 Shell and sum.py file. The shell shows the execution of the script, which prompts for two numbers and prints their sum. The inputs are 5.5 and 1.3, and the output is 6.8. The shell then prompts for a number again, and the user enters 5,5. A red box highlights the input line, and an arrow points to the input with the text 'Спробуємо ввести число 5,5 через кому, а не крапку' (Let's try to enter the number 5,5 using a comma, not a period).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5.5
Введіть число:1.3
5.5 + 1.3 = 6.8
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5,5
```

```
a=float(input("Введіть число:"))
b=float(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```

Спробуємо ввести число 5,5 через кому, а не крапку.

```
Python 3.6.5 Shell
File Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5.5
Введіть число:1.3
5.5 + 1.3 = 6.8
>>>
==== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py ====
Введіть число:5,5
Traceback (most recent call last):
  File "C:\Users\User\AppData\Local\Programs\Python\Python36\sum.py", line 1, in <module>
    a=float(input("Введіть число:"))
ValueError: could not convert string to float: '5,5'
>>>
```

```
sum.py - C:\Users\User\AppData\Local\Programs\Python\Python3...
File Edit Format Run Options Window Help
a=float(input("Введіть число:"))
b=float(input("Введіть число:"))
c=a+b
print(a,"+",b,"=",c)
```

Python розуміє розділювач дробової частини лише у вигляді крапки!

В результаті отримаємо помилку.

Кома в написанні дробового числа

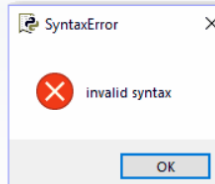
```
Введіть число:5,5
ValueError: could not convert string to float: '5,5'
```

Правильне написання

5.5

Ім'я змінної, що починається з цифри

```
2second=float(input("Введіть число:"))
```



second2

Написання імені змінної один раз з великої літери, інший – з малої

```
Second=float(input("Введіть число:"))
sum=first+second
NameError: name 'second' is not defined
```

Ось кілька типових помилок, яких потрібно уникати!

Здається, що Python має занадто багато обмежень і постійно вимагає якихось особливих дій, невластивих для звичайної людини, але крім обмежень є й додаткові можливості.

Наприклад, виявляється, що в Python можна множити текст.

```
>>> print("мама"*5)  
мамамамамамамамама
```



Отже, ми можемо взяти текст і помножити його на якийсь число. В результаті отримаємо відповідну кількість копій цього тексту.

З клавіатури завжди вводимо текст

- ♦ "7" – це не число, а текст

Можна перетворити текст на число

- ♦ `int("7")` – це число 7

Дробове число пишемо лише через крапку

- ♦ 5.5 – це число
- ♦ 5,5 – це помилка

Можна додавати і множити текст

- ♦ `"5"+"4"="54"`
- ♦ `"мама"*2="мамамама"`



Python має багато особливостей, про які варто пам'ятати!

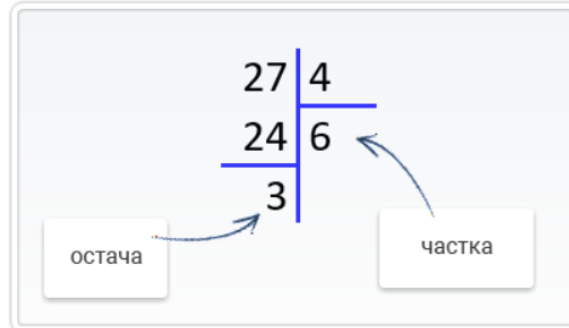


Ще дві арифметичні операції Python ви могли не зустрічати в математиці, але в інформатиці їх використовують досить часто.

Операції, пов'язані з діленням

// – частка від ділення

% – остача від ділення



Наприклад, $27 // 4 == 6$, $8 // 3 == 2$, $8 // 2 == 4$,
 $27 \% 4 == 3$, $8 \% 3 == 2$, $8 \% 2 == 0$.

Вправа

Вправа № 1



Вправа 1. Створи програму розв'язання задачі в середовищі Python.

Задача

- На 1 м^2 площі клумби, що має довжину **a** та ширину **b**, припадає 4 саджанці.
- Потрібно порахувати довжину огорожі та кількість саджанців, які треба закупити.

Вхідні дані		Вихідні дані	
Довжина	Ширина	Огорожа	Саджанці
2	3	10	24
5	25	60	500
1	8	18	32
123	654	1554	321768



Розв'яжи задачу в середовищі Python та перевір правильність її виконання за даними, наведеними в таблиці.

Вправа № 2



Вправа 2. Створи програму розв'язання задачі в середовищі Python.

Задача

- ♦ Додатково спробуй порахувати довжину огорожі та кількість саджанців для круглої клумби, радіус якої вводить користувач.



Розв'яжи задачу в середовищі Python.

Вивчення нового матеріалу

Слайд № 33

Задача

Левко сів виконувати домашнє завдання і просидів за столом 2 години.

З них **x** хвилин він чухав потилицю і дивився у вікно, **z** – шукав у письмовому столі гумку, щоб стерти у підручнику з англійської мови карикатуру на свого товариша, на малювання якої він витратив перед цим **y** хвилин.

Решту часу Левко перекладав англійські слова. Скільки слів він встиг перекласти, якщо переклад одного слова у нього займав 5 хвилин?



Розв'яжемо дуже практичну задачу.

Слайд № 34

Вихідними даними, які ми очікуємо побачити в результаті, буде відповідь на поставлене питання. Тобто кількість слів, які учень переклав.

Вхідні дані

Вводимо з клавіатури три цілих числа **x, y, z**.

Вихідні дані

На екран виводиться одне число – шукана кількість слів.



Визначимо вхідні та вихідні дані.

Слайд № 35

Розглянемо алгоритм розв'язання задачі на прикладі деяких вхідних даних.

Вхідні дані

Вводимо з клавіатури три цілих числа **x, y, z**.

Вихідні дані

На екран виводиться одне число – шукана кількість слів.

Приклад вхідних і вихідних даних

Вхід: 10 25 8

Вихід: 15

2 години = 120 хвилин

$120 - (10 + 25 + 8) = 77$ (хв) -
час на переклад слів

$77 : 5 = 15,4$ (сл.) -
кількість перекладених слів

Оскільки перекласти можна
тільки цілу кількість слів,
відповідь буде - **15 слів**

Слайд № 36

Ось позначення і формули, необхідні
для розв'язання задачі.

Програма "Учень"

- ♦ **x** – потилиця і вікно
- ♦ **y** – малювання
- ♦ **z** – пошук гумки
- ♦ **k** – час на переклад
 $k = 120 - (x + y + z)$
- ♦ **s** – кількість слів
- ♦ на одне слово 5 хвилин
- ♦ $s = k // 5$



Тепер ти можеш виконати вправу!

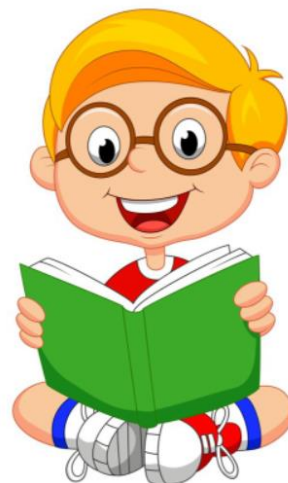
Вправа № 3

**Вправа 3.** Створи програму розв'язання задачі в середовищі Python.**Задача**

Левко сів виконувати домашнє завдання і просидів за столом 2 години.

З них **x** хвилин він чухав потилицю і дивився у вікно, **z** – шукав у письмовому столі гумку, щоб стерти у підручнику з англійської мови карикатуру на свого товариша, на малювання якої він витратив перед цим **y** хвилин.

Решту часу Левко перекладав англійські слова. Скільки слів він встиг перекласти, якщо переклад одного слова у нього займав 5 хвилин?



Розв'яжи задачу в середовищі Python.

Вивчення нового матеріалу

Слайд № 37

```
Python 3.6.5 Shell
File Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

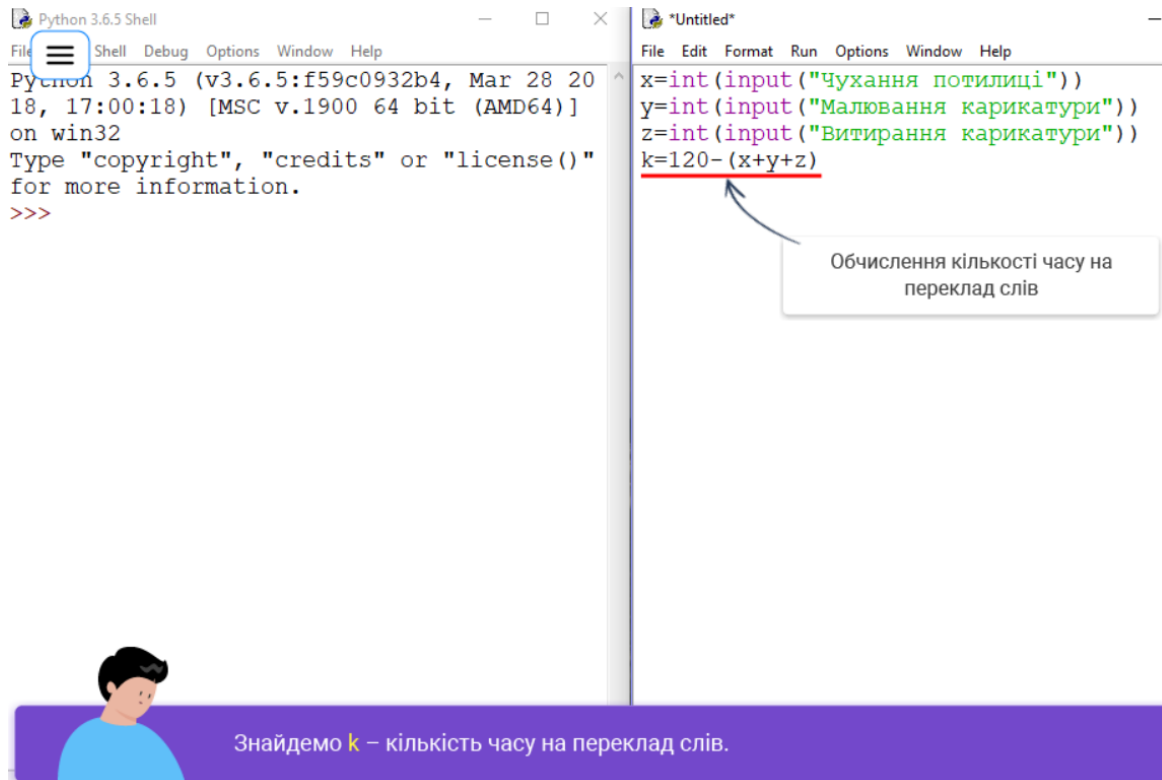
Untitled
File Edit Format Run Options Window Help
x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
```

Вхідні дані



Напишемо програму. Спочатку введемо дані.

Слайд № 38



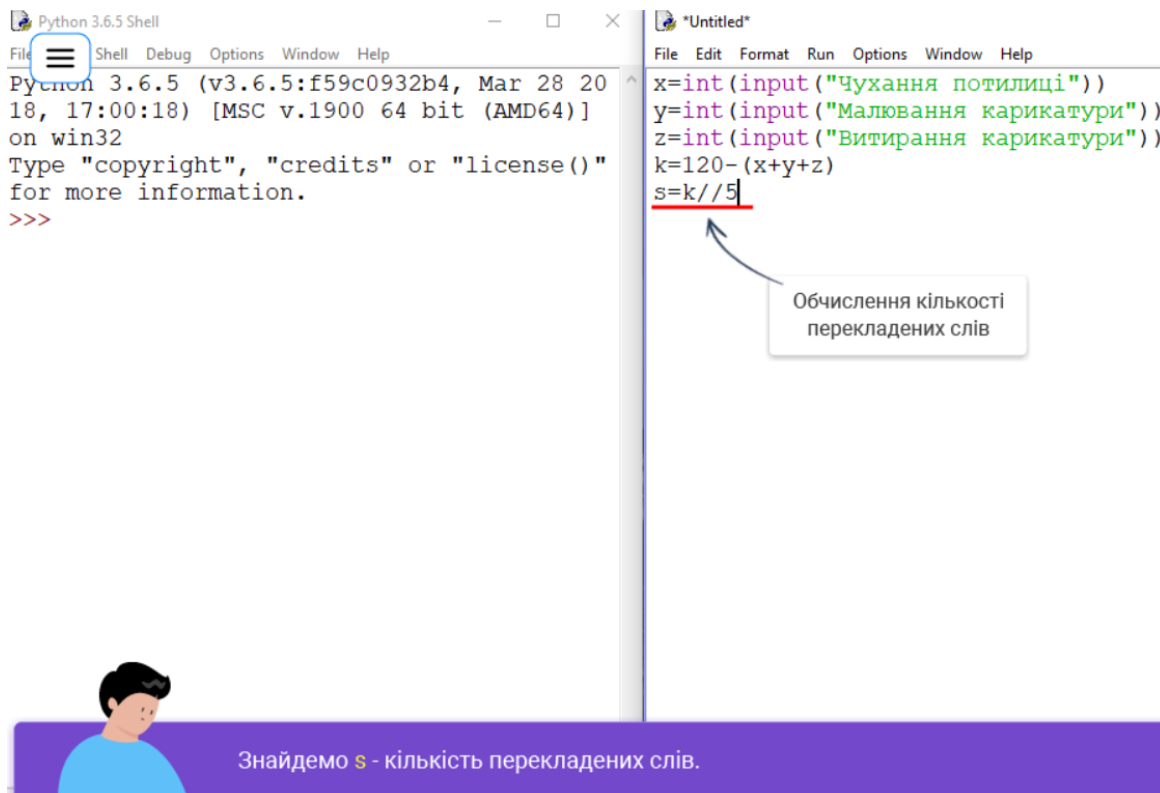
```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
k=120-(x+y+z)
```

Обчислення кількості часу на переклад слів

Знайдемо **k** – кількість часу на переклад слів.

Слайд № 39



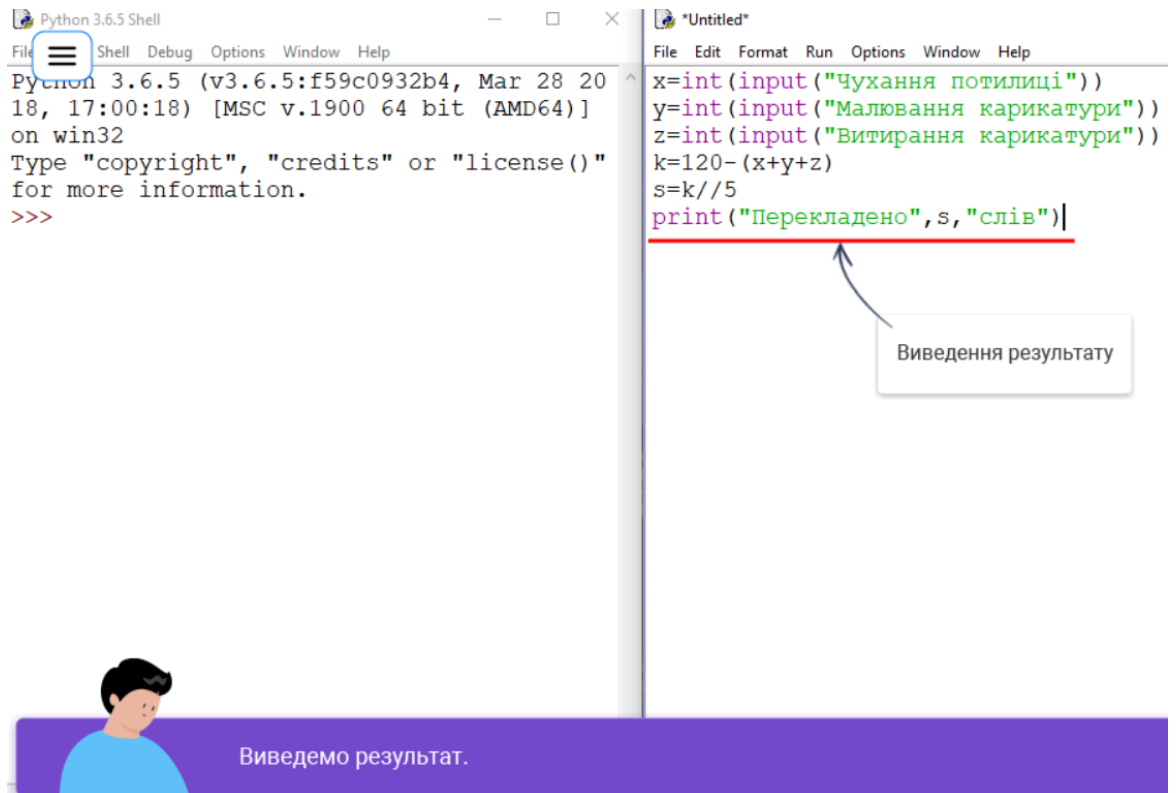
```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
k=120-(x+y+z)
s=k//5
```

Обчислення кількості перекладених слів

Знайдемо **s** - кількість перекладених слів.

Слайд № 40



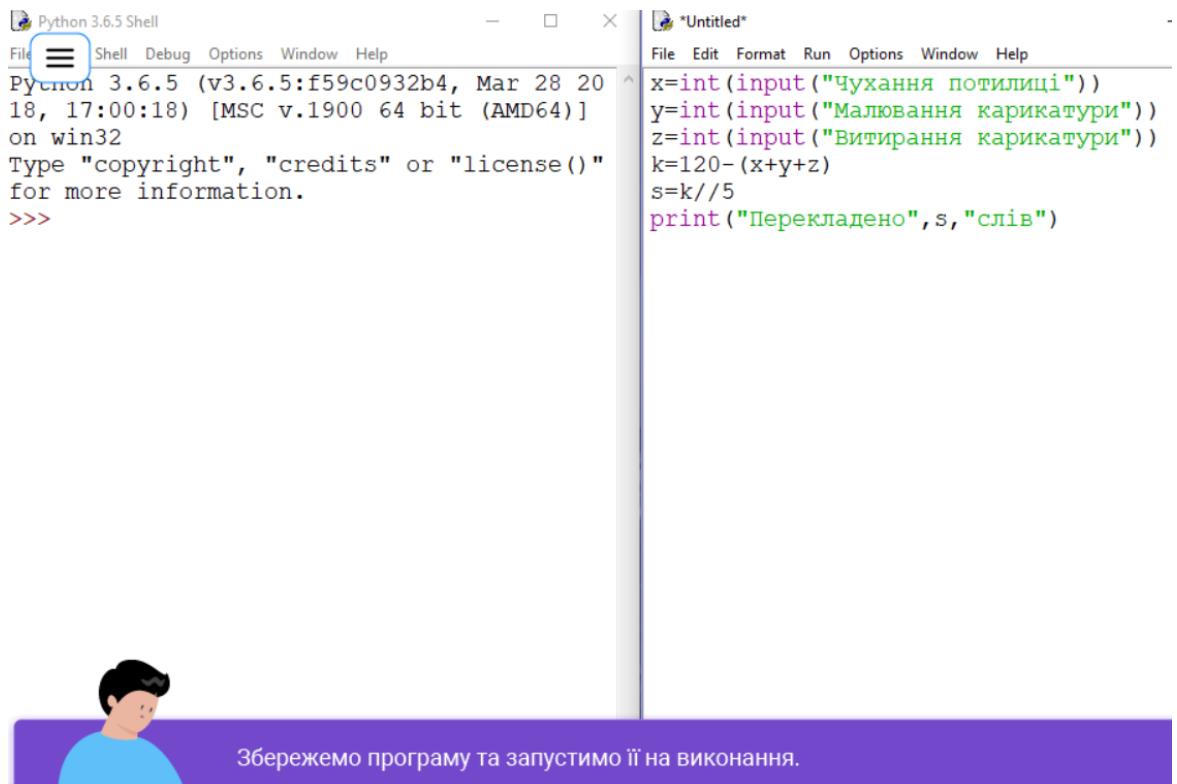
The screenshot shows two windows. The left window is the Python 3.6.5 Shell, displaying the version and system information. The right window is an Untitled text editor containing the following Python code:

```
x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
k=120-(x+y+z)
s=k//5
print("Перекладено", s, "слів")
```

An arrow points from a text box labeled "Виведення результату" (Output result) to the `print` statement in the code.

Виведемо результат.

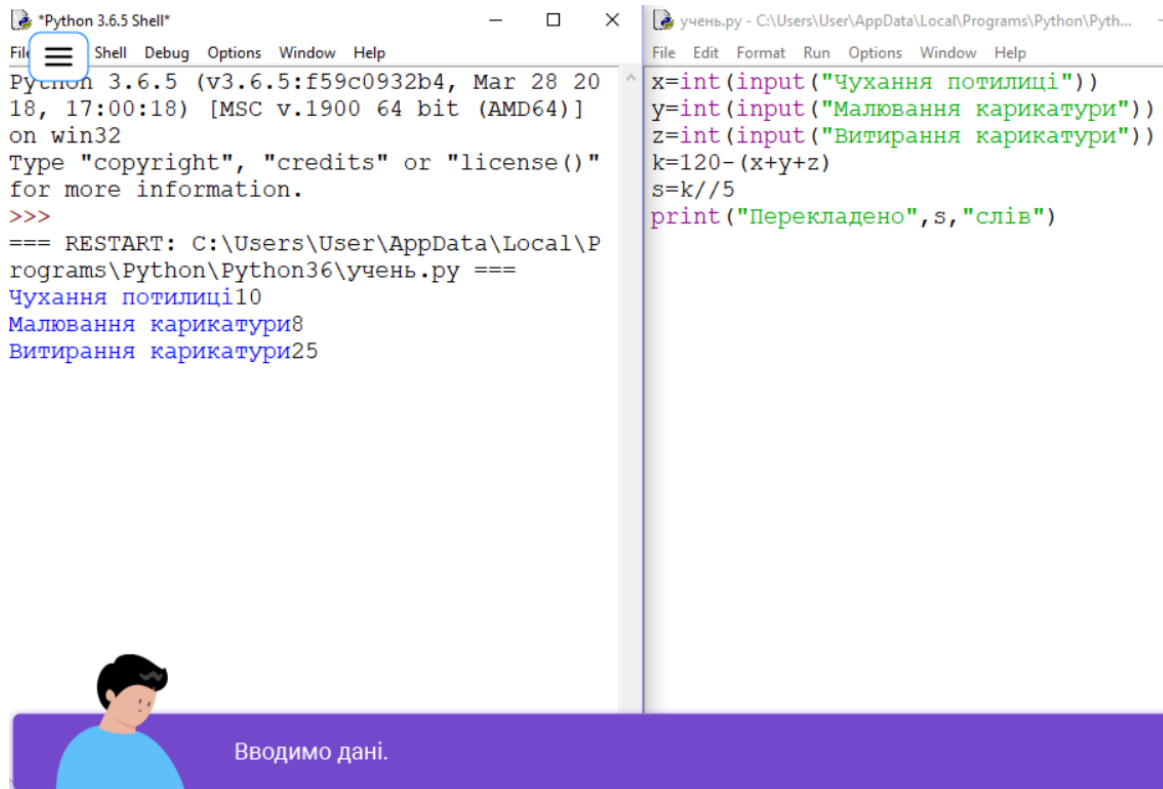
Слайд № 41



The screenshot shows the same two windows as Slide 40. The code in the Untitled text editor is identical to the one in Slide 40.

Збережемо програму та запустимо її на виконання.

Слайд № 42



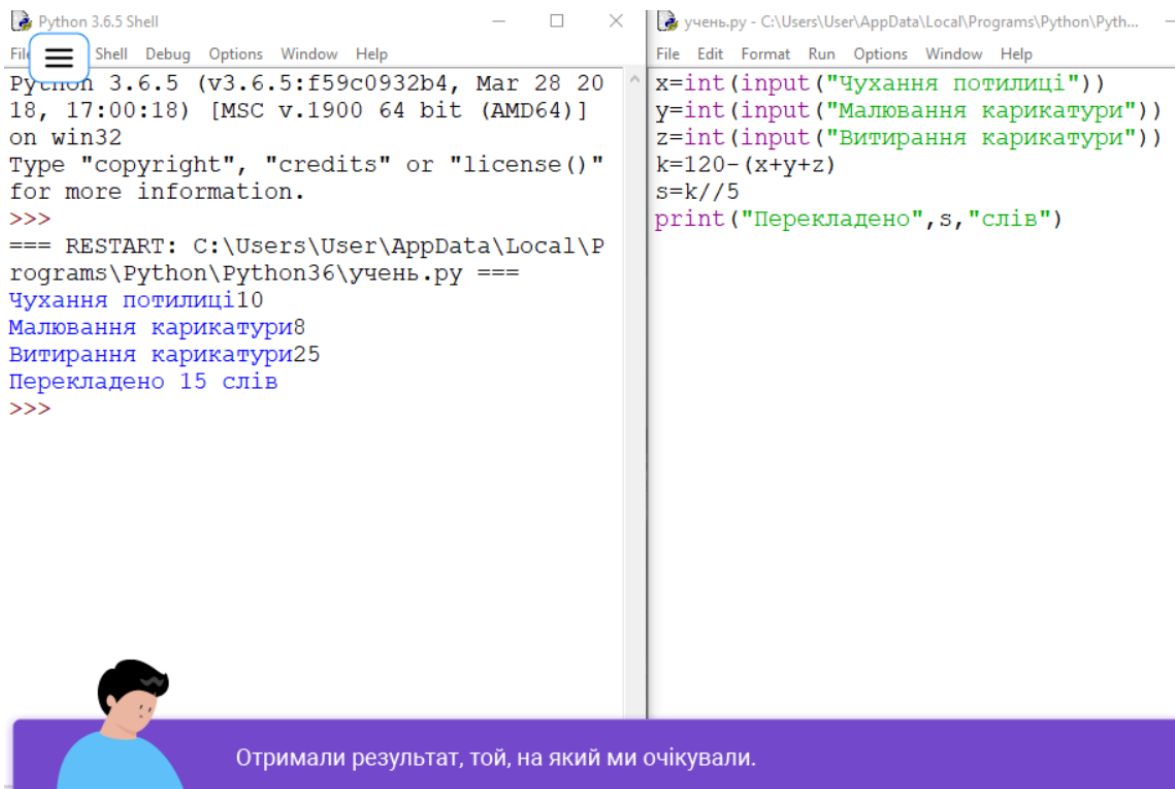
The screenshot shows two windows. The left window is the Python 3.6.5 Shell, displaying the prompt and the execution of the script. The right window shows the source code of the script 'учень.py'.

```
Python 3.6.5 Shell
File Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\учень.py ===
Чухання потилиці10
Малювання карикатури8
Витирання карикатури25
```

```
учень.py - C:\Users\User\AppData\Local\Programs\Python\Python36\учень.py
File Edit Format Run Options Window Help
x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
k=120-(x+y+z)
s=k//5
print("Перекладено",s,"слів")
```

Вводимо дані.

Слайд № 43



The screenshot shows the same two windows as Slide 42, but the shell window now displays the final output of the script, including the calculated number of words.

```
Python 3.6.5 Shell
File Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\User\AppData\Local\Programs\Python\Python36\учень.py ===
Чухання потилиці10
Малювання карикатури8
Витирання карикатури25
Перекладено 15 слів
>>>
```

```
учень.py - C:\Users\User\AppData\Local\Programs\Python\Python36\учень.py
File Edit Format Run Options Window Help
x=int(input("Чухання потилиці"))
y=int(input("Малювання карикатури"))
z=int(input("Витирання карикатури"))
k=120-(x+y+z)
s=k//5
print("Перекладено",s,"слів")
```

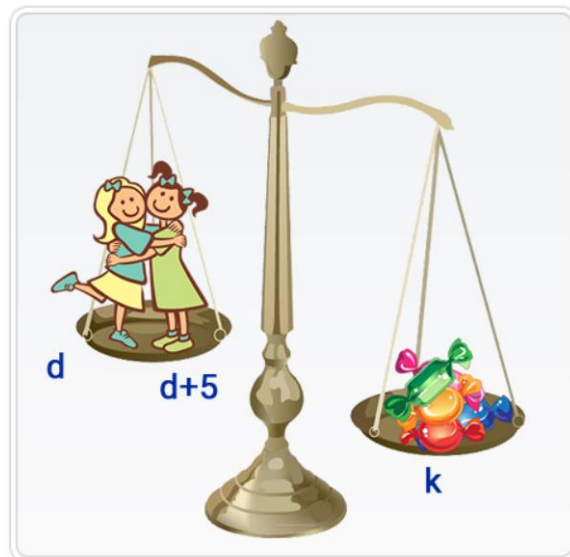
Отримали результат, той, на який ми очікували.

Слайд № 44

Задача

Якщо на шальку терезів посадити Даринку, яка важить d кг і Наталю, яка важить на 5 кг більше, а на іншу шальку насипати k кг цукерок, то скільки цукерок потрібно з'їсти дівчаткам, щоб шальки терезів врівноважилися?

Зауваження. Вважаємо, що цукерок насипали стільки, що вони переважили дівчат.



Ще одна задача, нехай не дуже практична, але смачна.

Слайд № 45

Розглянемо алгоритм розв'язання задачі на прикладі деяких вхідних даних

Вхідні дані

Вводимо з клавіатури два цілих числа d і k .

Вихідні дані

На екран виводиться одне число (s).

Приклад вхідних і вихідних даних

Вхід: 40 88

Вихід: 1,5

$40 + 5 = 45$ (кг) - вага Наталі

$40 + 45 = 85$ (кг) -
вага дівчаток

$88 - 85 = 3$ (кг) - різниця між
шальками терезів

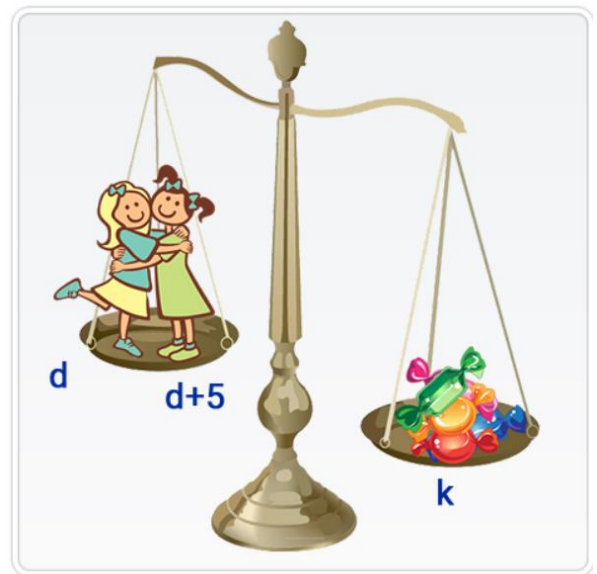
$3 / 2 = 1,5$ (кг) - потрібно
з'їсти дівчаткам для
урівноваження терезів



Визначимо вхідні та вихідні дані.

Програма "Дівчата та цукерки"

- ♦ **d** – вага Даринки
- ♦ **k** – вага цукерок
- ♦ **girls** – загальна вага дівчаток
 $girls = d + d + 5$
- ♦ **s** – вага цукерок, яку мають з'їсти дівчата
- ♦ $s = (k - girls) / 2$



Отже, нами сформульовано математичні формули розв'язання задачі. Тепер ти зможеш створити програму в середовищі Python.

Вправа

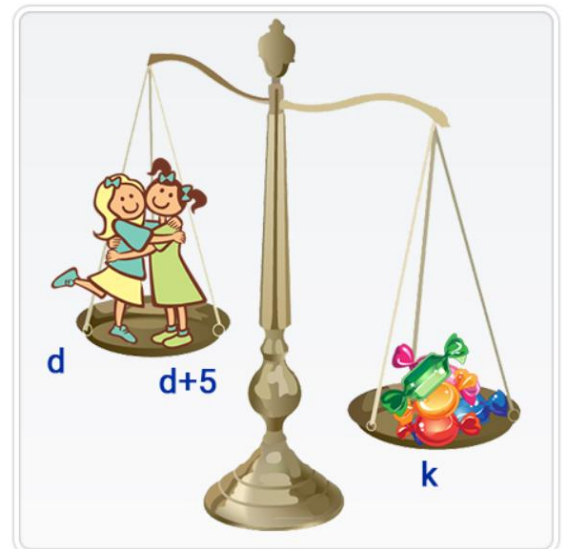
Вправа № 4



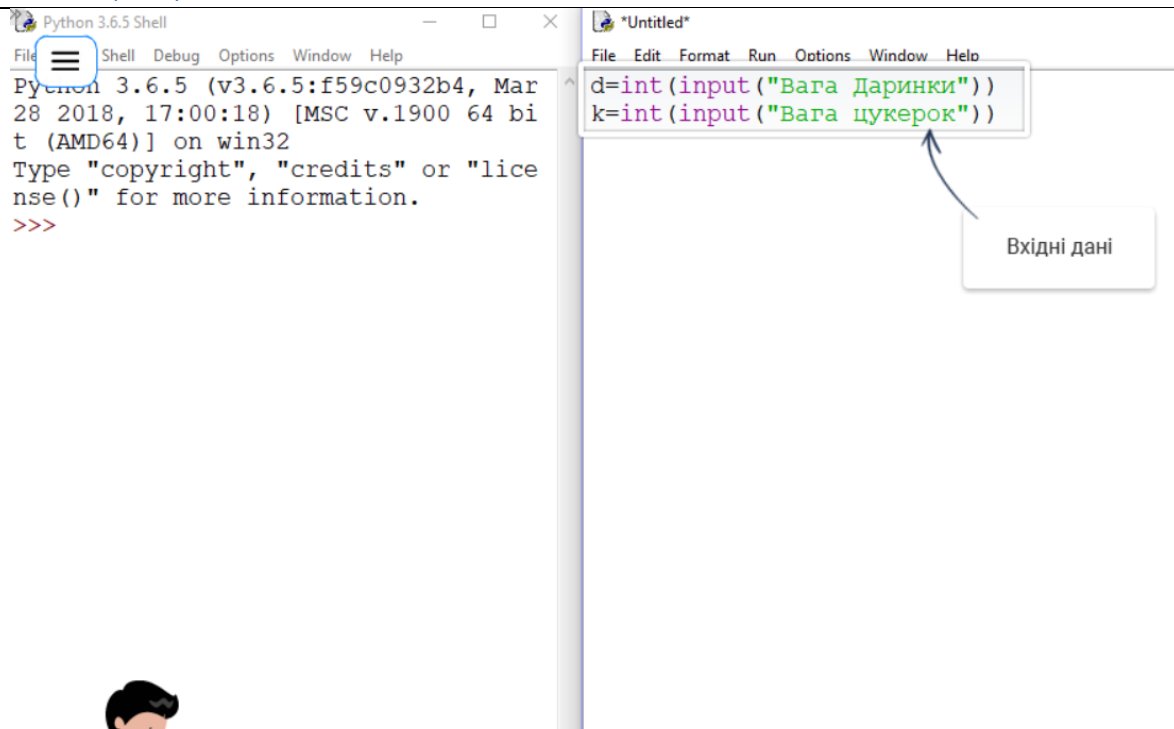
Вправа 4. Створи програму розв'язання задачі в середовищі Python.

Задача

Якщо на шальку терезів посадити Даринку, яка важить **d** кг, і Наталю, яка важить на 5 кг більше, а на іншу шальку насипати **k** кг цукерок, то скільки цукерок потрібно з'їсти дівчаткам, щоб шальки терезів врівноважилися?



Слайд № 47



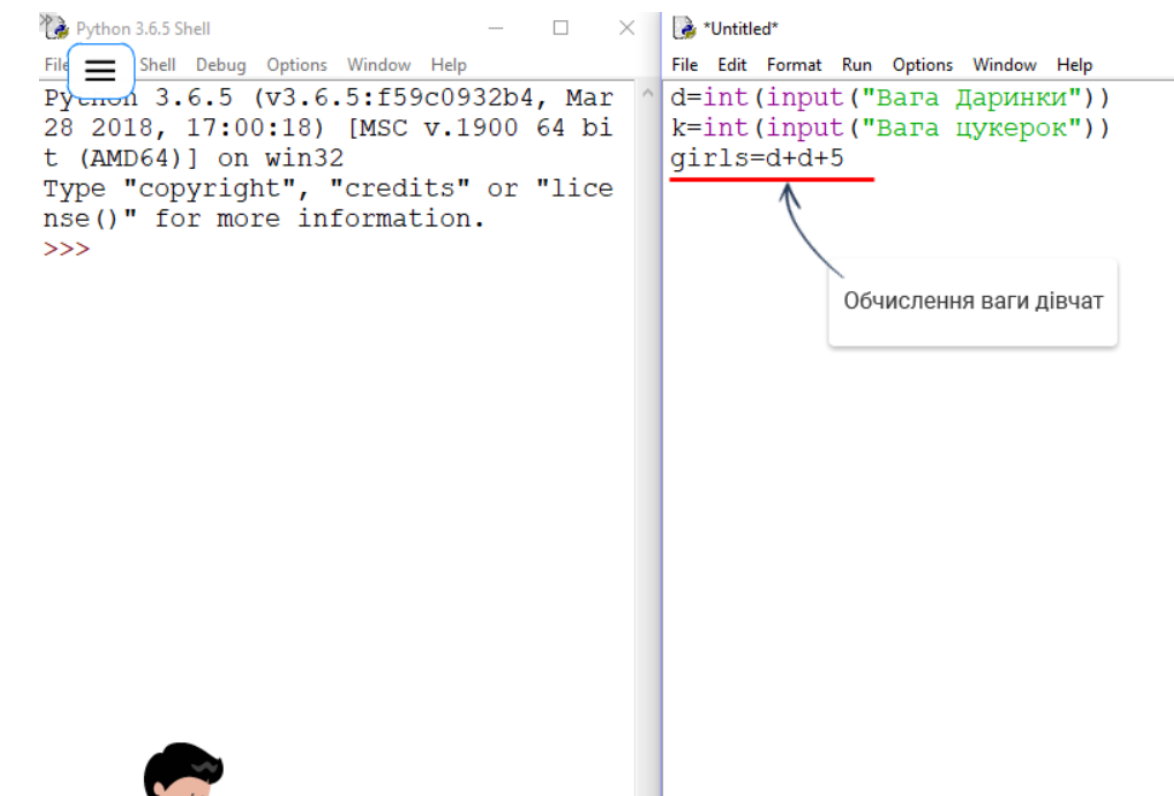
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

```
d=int(input("Вага Даринки"))  
k=int(input("Вага цукерок"))
```

Вхідні дані

У програмі знову починаємо з вхідних даних. Це вага Даринки і цукерок.

Слайд № 48



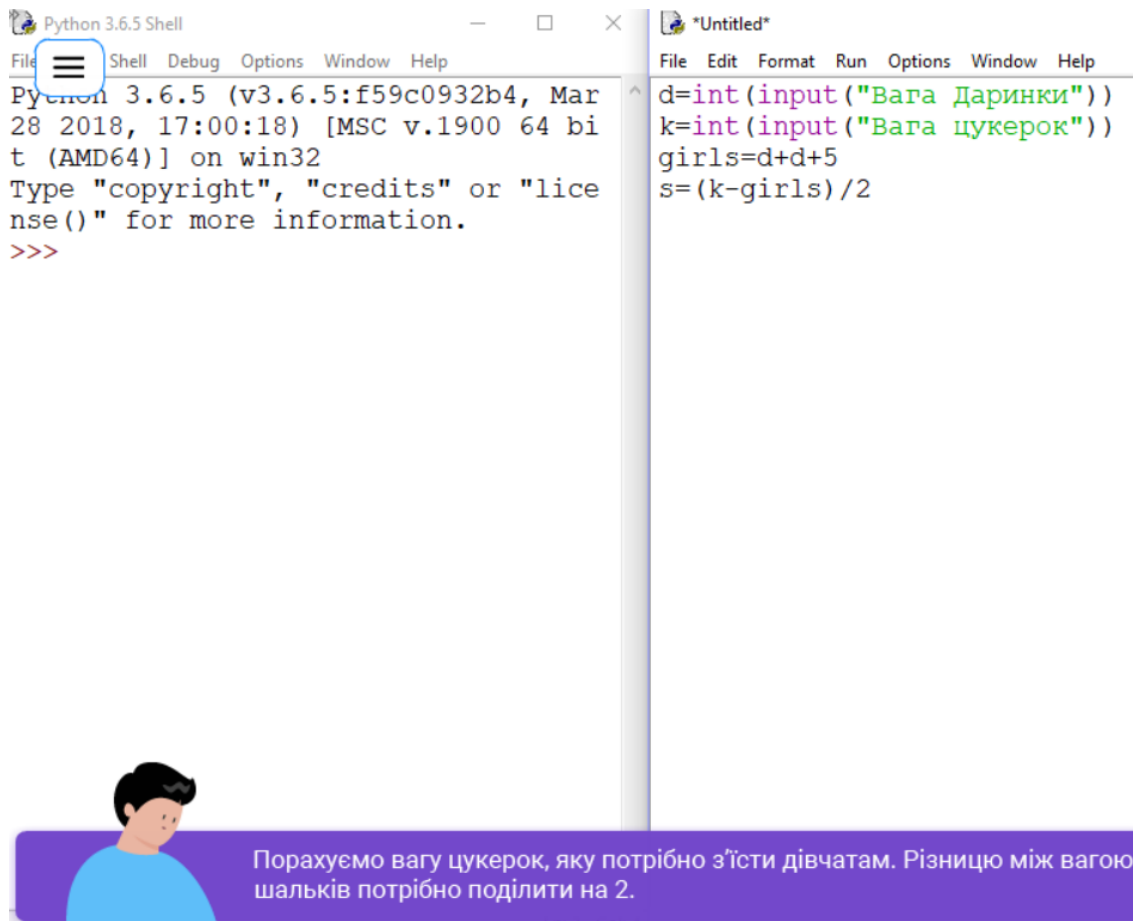
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

```
d=int(input("Вага Даринки"))  
k=int(input("Вага цукерок"))  
girls=d+d+5
```

Обчислення ваги дівчат

Обчислимо вагу шальки терезів з дівчатами присвоїмо її змінній **girls**.

Слайд № 49

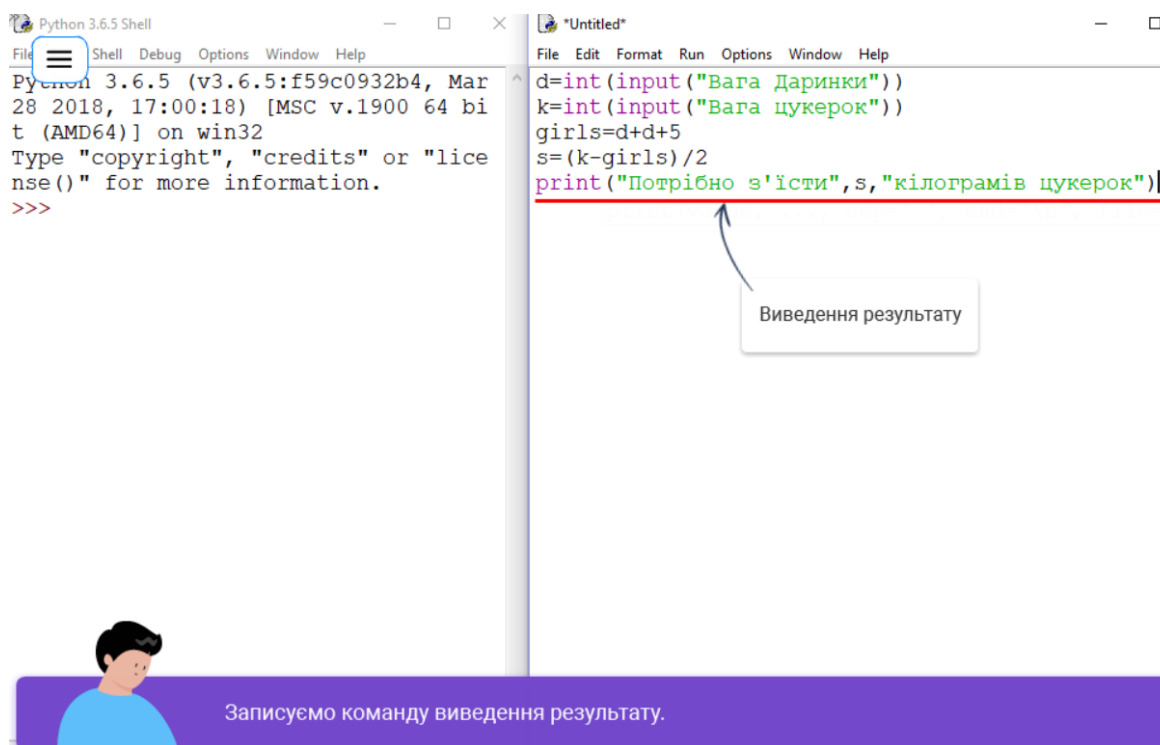


Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

```
d=int(input("Вага Даринки"))  
k=int(input("Вага цукерок"))  
girls=d+d+5  
s=(k-girls)/2
```

Порахуємо вагу цукерок, яку потрібно з'їсти дівчатам. Різницю між вагою шальків потрібно поділити на 2.

Слайд № 50



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

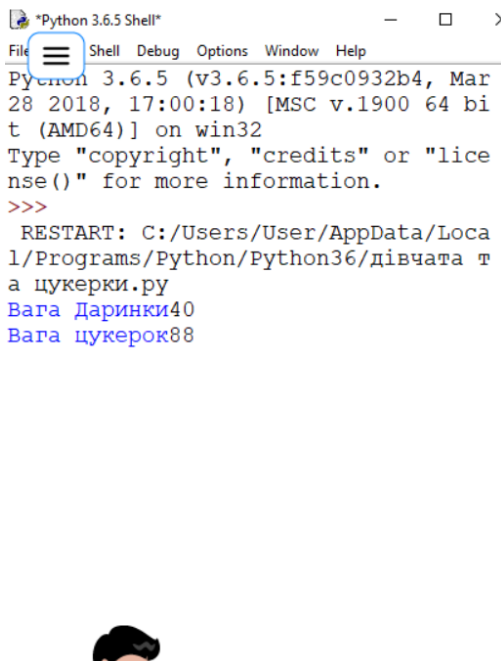
```
d=int(input("Вага Даринки"))  
k=int(input("Вага цукерок"))  
girls=d+d+5  
s=(k-girls)/2  
print("Потрібно з'їсти",s,"кілограмів цукерок")
```

Виведення результату

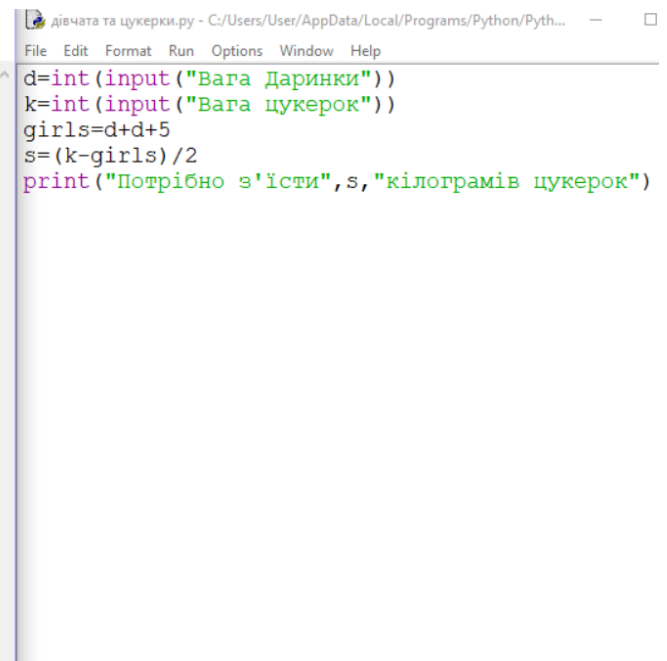
Записуємо команду виведення результату.

Збережемо програму та запустимо її на виконання.

Слайд № 51



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/User/AppData/Local/Programs/Python/Python36/дівчата та цукерки.py
Вага Даринки40
Вага цукерок88

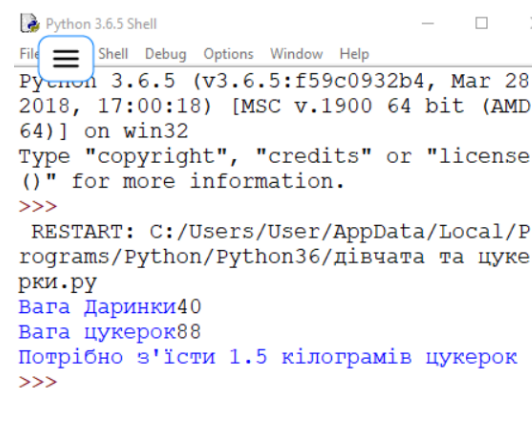


```
d=int(input("Вага Даринки"))
k=int(input("Вага цукерок"))
girls=d+d+5
s=(k-girls)/2
print("Потрібно з'їсти",s,"кілограмів цукерок")
```

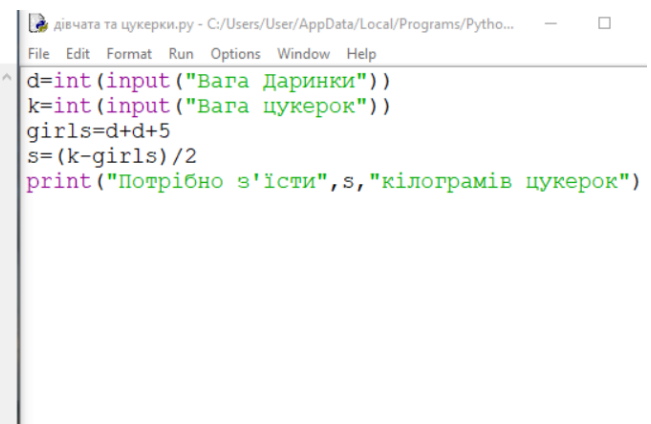


Перевіряємо програму для вже відомого результату, щоб переконатись, що вона працює правильно.

Слайд № 52



Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/User/AppData/Local/Programs/Python/Python36/дівчата та цукерки.py
Вага Даринки40
Вага цукерок88
Потрібно з'їсти 1.5 кілограмів цукерок
>>>



```
d=int(input("Вага Даринки"))
k=int(input("Вага цукерок"))
girls=d+d+5
s=(k-girls)/2
print("Потрібно з'їсти",s,"кілограмів цукерок")
```

Різниця між вагою цукерок і дівчат становить
 $88 - (40 + 40 + 5) = 3 \text{ кг}$



Якщо дівчата з'їдять 1,5 кг цукерок, то вони стануть на 1,5 кг важчими, а цукерки — на 1,5 кг легшими. І терези врівноважаться.

Домашнє
завдання



Домашнє завдання.

Задача

Є **n** білочок та **m** горішків, які потрібно розділити порівну між білочками. По скільки цілих горіхів отримає кожна білочка і скільки горіхів залишиться?



Створи у Python програму, що розв'язує цю задачу.