

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій

Кафедра комп'ютерної інженерії

Методичні вказівки до виконання лабораторних робіт

з дисципліни

**«Micropython для одноплатних
мікрокомп'ютерів»**

за освітньо-професійною програмою «Комп'ютерна інженерія» для
здобувачів першого (бакалаврського) рівня вищої освіти спеціальності

123 – Комп'ютерна інженерія

Тернопіль – 2023

Методичні вказівки до виконання лабораторних робіт з дисципліни «Micropython для одноплатних мікрокомп'ютерів» для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 – Комп'ютерна інженерія / Батько Ю.М. Тернопіль: ЗУНУ, 2023. 32 с.

Автори:

Батько Ю.М., кандидат технічних наук, доцент кафедри комп'ютерної інженерії.

Рецензенти:

Биковий Павло Євгенович, кандидат технічних наук, доцент кафедри інформаційно-обчислювальної систем і управління Західноукраїнського національного університету

Івасьєв Степан Володимирович кандидат технічних наук, доцент, доцент кафедри кібербезпеки Західноукраїнського національного університету

Рекомендовано на засіданні кафедри комп'ютерної інженерії ЗУНУ
(Протокол № 6 від 23.02.2022 р.)

ЗМІСТ

Вступ	4
Лабораторна робота №1. Тема: Основні поняття та принципи побудови програм на мові python.	5
Лабораторна робота №2. Математичні операції та алгоритми розгалуження на мові python	12
Лабораторна робота №3. Тема: Циклічні алгоритми та масиви даних .	17
Лабораторна робота №4. Одноплатні комп'ютери Raspberry Pi	26
РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ	32

Вступ

Python - інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Високорівневі вбудовані структури даних у поєднанні з динамічним набором тексту та динамічним зв'язуванням роблять її дуже привабливою для швидкого програмування додатків, а також для використання як мови сценаріїв або для з'єднання існуючих компонентів разом. Простий, легкий у вивченні синтаксис Python підкреслює читабельність, а отже, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній чи двійковій формі безкоштовно за всіма основними платформами, і їх можна вільно поширювати.

Часто програмісти використовують Python через підвищену продуктивність, яку він забезпечує. Оскільки кроку компіляції немає, цикл редагування-тестування-налагодження надзвичайно швидкий. Налагодження програм Python легко: помилка або неправильний вхід ніколи не спричинить помилку сегментації. Натомість, коли перекладач виявляє помилку, він створює виняток. Коли програма не вловлює виняток, інтерпретатор виводить слід стека. Налагоджувач рівня джерела дозволяє перевіряти локальні та глобальні змінні, оцінювати довільні вирази, встановлювати точки прориву, переходити через код рядок за часом тощо.

Серед недоліків Python можна зазначити те, що код написаний для однієї з версій Python не завжди правильно може виконуватись для іншої версії, або ж взагалі не буде працювати.

Лабораторна робота №1. Тема: Основні поняття та принципи побудови програм на мові python.

Мета: Отримати теоретичні знання та практичні навички по розробці програмних додатків на мові програмування python.

Завдання: Встановити, налаштувати середовище розробки мови Python та виконати просту програму написану на мові Python

Теоретичні відомості.

В комплекті разом з інтерпретатором Python йде IDLE (інтегроване середовище розробки). За своєю суттю вона подібна до інтерпретатора, запущеного в інтерактивному режимі з розширеним набором можливостей (підсвічування синтаксису, перегляд об'єктів, налагодження тощо). Для запуску програм на Python необхідно встановити на робочу станцію інтерпретатор даної мови. Актуальну версію інтерпретатора можна скачати на офіційному сайті <https://www.python.org/downloads/>. Після чого пройти процедуру інсталяції.

IDLE в Windows необхідно перейти в папку Python в меню "Пуск" і знайти там ярлик з ім'ям "IDLE (Python 3.X XX-bit)".

Проте на сьогоднішній момент існує безліч версій програмних продуктів, що дозволяють писати програмні додатки у більш зручних редакторах, одним з таких є редактор коду Atom.

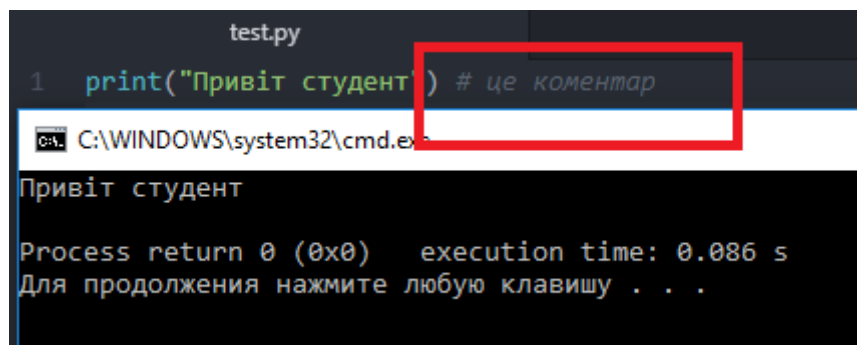
Будь-яка Python-програма складається з послідовності допустимих символів, записаних в певному порядку і за певними правилами.

Програма включає в себе:

- коментарі;
- команди;
- знаки пунктуації;
- ідентифікатори;

- ключові слова.

Коментарі в Python позначаються символом #, що розташовується перед ним і тривають до кінця рядка (тобто, в Python всі коментарі є однорядковими), при цьому не допускається використання перед символом # лапок.



```
test.py
1 print("Привіт студент") # це коментар
C:\WINDOWS\system32\cmd.exe
Привіт студент
Process return 0 (0x0) execution time: 0.086 s
Для продовження натисніть будь-яку клавішу . . .
```

Ідентифікатори в Python це імена використовуються для позначення змінної, функції, класу, модуля або іншого об'єкта. Ідентифікаторами **не можуть** бути зарезервовані слова які є керуючими конструкціями для мови Python:

['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Типи даних, що використовуються в Python.

- None (невизначене значення змінної)
- Логічні змінні (Boolean Type)
- Числа (Numeric Type)
 - int - ціле число
 - float - число з плаваючою точкою
 - complex - комплексне число
- Списки (Sequence Type)
 - list - список
 - tuple - кортеж
 - range - діапазон

– Рядки (Text Sequence Type)

- str

Введення та виведення даних.

Введення даних здійснюється за допомогою команди:

input (список введення)

```
a = input ()
```

У дужках функції можна вказати повідомлення - коментар до вводиться даними:

```
a = input ( "Введіть кількість:")
```

Команда `input ()` за замовчуванням сприймає вхідні дані як **рядок символів**. Тому, щоб змінити тип даних перед командою прописується той тип даних який необхідний, наприклад щоб ввести цілочисельне значення, слід вказати тип даних `int ()`:

```
a = int (input ())
```

Вивід інформації здійснюється за допомогою команди

print (список виведення)

```
1 a=input('Введіть значення змінної a =')
2 b=input('Введіть значення змінної b =')
3 #інтерпретатор сприймає введеної інформацію як стрічкову
4 # кожне зі значень виводиться окремо
5 print('Сума a+b=')
6 print(a+b)
7 # декілька значень виводять в одній команді
8 print('Сума a+b=',a+b)
9 a=int (input('Введіть значення змінної a ='))
10 b=int (input('Введіть значення змінної b ='))
11 #інтерпретатор сприймає введеної інформацію як числову
12 # кожне зі значень виводиться окремо
13 print('Сума a+b=')
14 print(a+b)
15 # декілька значень виводять в одній команді
16 print('Сума a+b=',a+b)
```

```
D:\Python_3_8_0\python.exe
Введіть значення змінної a =5
Введіть значення змінної b =6
Сума a+b=
56
Сума a+b= 56
Введіть значення змінної a =5
Введіть значення змінної b =6
Сума a+b=
11
Сума a+b= 11
```

Математичні операції на Python.


Нижче наведена таблиця, можливих математичних операцій.

назва	команда	приклад	результат
+	додавання	2+1	3
-	віднімання	7-3	4
*	множення	2*3	6
/	ділення	4/2	2
//	Цілочисельне ділення	5/3	1
%	залишок від ділення	5%2	1
**	піднесення до степеня	2**3	8
-x	Зміна знаку числа x		

Для Python характерна наступна послідовність виконання математичних операцій.

- дужки ();
- піднесення до степеня **;
- множення *, ділення /, остача від ділення %;
- додавання +, віднімання -;

```
1 print(2*4) # множення
2 print(5//3) # цілочисельне ділення
3 a=3 # призвоєння значення змінній
4 print(a) # вивід змінної
5 print(-a) # зміна знаку змінної
```



D:\Python_3_8_0\python.exe

8
1
3
-3

Індивідуальні завдання.

Варіанти індивідуального завдання, для кожного варіанта, обираються за формулою

$(\text{№}_\text{ студента}_\text{ в}_\text{ журналі} \% \text{ кількість}_\text{ варіантів}_\text{ завдання}) + 1$

Завдання 1

Номер варіанту	Завдання
1	<p>Ввести з клавіатури:</p> <ul style="list-style-type: none"> - ім'я ("Ваше, ім'я?") - дата народження ("Ваша дата народження?") - освіту ("Де Ви вчитеся?") <p>Після цього вивести в три рядки:</p> <p>"Ваше ім'я"</p> <p>"Дата народження"</p> <p>"Ви вчитеся в"</p>
2	<p>Ввести з клавіатури:</p> <ul style="list-style-type: none"> - Прізвище, ("Ваше прізвище?") - місце народження ("Де Ви народилися?") - музика ("Яка музика подобається?") <p>Після цього вивести в три рядки:</p> <p>"Ваші ім'я, прізвище"</p> <p>"Ви народилися в"</p> <p>"Ваша улюблена музика"</p>
3	<p>Ввести з клавіатури:</p> <ul style="list-style-type: none"> - ПІБ (наприклад, "Ваші прізвище, ім'я, по батькові?") - вік ("Скільки Вам років?") - місце проживання ("Де Ви живете?") <p>Після цього вивести в три рядки:</p> <p>"Ваші ім'я, прізвище, по батькові"</p> <p>"Ваш вік"</p> <p>"Ви живете в"</p>
4	<p>Ввести з клавіатури:</p> <ul style="list-style-type: none"> - Прізвище, ім'я ("Ваші прізвище, ім'я?") - освіту ("На якому курсі Ви вчитеся?") - ПІБ Вашого куратора ("ПІБ Вашого куратора?") <p>Після цього вивести в три рядки:</p> <p>"Ваші ім'я, прізвище"</p>

	<i>"Ви вчитесь на :"+x+"курсі" "ПІБ Вашого куратора ""</i>
5	Вести з клавіатури: - Прізвище, ім'я ("Ваші прізвище, ім'я?") - вік ("Скільки Вам років?") - номер телефону ("Номер Вашого телефону?") Після цього вивести в три рядки: <i>"Ваші ім'я, прізвище" "Ваш вік" "Ваш номер телефону"</i>

Завдання 3

Номер варіанту	Завдання
1	Користувач вводить рік свого народження, обчислити скільки йому років
2	Користувач вводить поточний місяць, визначити скільки місяців до кінця року
3	Користувач вводить вартість покупки та наявна у нього сума грошей, визначити скільки залишиться коштів у покупця після оплати покупки
4	Користувач вводить кількість під'їздів в будинку, кількість поверхів та кількість квартир на сходовій площадці, визначити скільки квартир є в будинку
5	Користувач вводить кількість наявних у нього монет вартістю 1, 2, 5, 10, 25, 50, визначити скільки всього коштів є у користувача, при цьому відповідь вивести в форматі "x грн. у копійок"

Контрольні запитання.

- 1) Що таке Python і де його використовують?
- 2) Процес встановлення Python.
- 3) Переваги та недоліки використання Python.
- 4) Структура мови на Python.
- 5) Способи виконання програм на Python.
- 6) Функції вводу/виводу в Python, особливості застосування.

Структура звіту.

- 1) Титулка
- 2) Відповіді на теоретичні запитання
- 3) Лістинг реалізованих програм згідно варіанту
- 4) Скрін-шоти роботи реалізованих програм
- 5) Висновки

Лабораторна робота №2. Математичні операції та алгоритми розгалуження на мові python

Мета: Отримати практичні навички для розв'язання прикладних задач з елементами механізмів розгалуження та складних математичних виразів.

Завдання: Ознайомитись з основними операторами для проведення складних математичних розрахунків та операторами реалізації алгоритмів розгалуження на мові програмування Python

Теоретичні відомості.

Для проведення обчислень з дійсними числами мову Python містить багато додаткових функцій, зібраних в бібліотеку (модуль), яка називається `math`. Для використання цих функцій на початку програми необхідно підключити математичну бібліотеку.

Для підключення бібліотек використовується ключова команда

import назва бібліотеки

import math

Нище наведено ряд функцій, що реалізовані для обчислення математичних виразів

Функція	Опис
округлення	
<code>int(x)</code>	Округлює число в бік нуля. Це стандартна функція, для її використання не потрібно підключати модуль <code>math</code> .
<code>round(x)</code>	Округлює число до найближчого цілого. Якщо дробова частина числа дорівнює 0.5, то число округляється до найближчого парного числа.
<code>round(x, n)</code>	Округлює число <code>x</code> до <code>n</code> знаків після коми. Це стандартна функція, для її використання не потрібно підключати модуль <code>math</code> .
<code>floor(x)</code>	Округлює число до меншого, при цьому <code>floor(1.5) == 1, floor(-1.5) == -2</code>

ceil(x)	Округлює число до більшого, при цьому ceil(1.5) == 2, ceil(-1.5) == -1
abs(x)	Модуль (абсолютна величина). Це - стандартна функція.
Коріння, логарифми	
sqrt(x)	Квадратний корінь. Використання: sqrt(x)
log(x)	Натуральний логарифм. При виклику у вигляді log(x, b) повертає логарифм за основою b.
e	Константа з значенням натуральних логарифмів $e = 2,71828 \dots$
тригонометрія	
sin(x)	Синус кута, що задається в радіанах
cos(x)	Косинус кута, що задається в радіанах
tan(x)	Тангенс кута, що задається в радіанах
asin(x)	Арксинус, повертає значення в радіанах
acos(x)	Арккосинус, повертає значення в радіанах
atan(x)	Арктангенс, повертає значення в радіанах
degrees(x)	Перетворює кут, заданий в радіанах, в градуси.
radians(x)	Перетворює кут, заданий в градусах, в радіани.
pi	Константа $\pi = 3.1415 \dots$

Індивідуальні завдання.

Варіанти індивідуального завдання, для кожного варіанта, обираються за формулою

$$(\text{№_студента_в_журналі \% кількість_варіантів_завдання}) + 1$$

Завдання 1

Номер варіанту	Завдання
1	<i>Розв'язати математичні вирази:</i>

	$x = 45 + 5 + (46 - 6)$ $y = \frac{17 - 4 * (18 / 6)}{21^2}$ $z = \frac{\cos(0.4) - \sin(2.4) + 45^{3+4}}{\pi^2}$
2	<p>Розв'язати математичні вирази:</p> $x = 34535 - 26236 + 4535 + (54 * 58)$ $y = \frac{1 + 7 - 66 * (0.6 / 0.6)}{(32 - 13)^2}$ $z = \frac{\sin(0.4) - \cos(2.4) + 4 - 5^{11-4}}{\pi^4 + e^3}$
3	<p>Розв'язати математичні вирази:</p> $x = 345 - 35 - 26 * 2 + 36 + 45 - 35 + (54 - 58)$ $y = \frac{71 - 6 + 6 * (4 - 9)}{(3 + 2 - 1 * 3)^2}$ $z = \frac{\text{tg}(0.14) - \cos(0.3) * 5^{0,5}}{e^3}$
4	<p>Розв'язати математичні вирази:</p> $x = (-35) * 26 + 45 - 35 + (5 * 5)$ $y = \frac{0.35 - 6 * (-0.1)}{(4532 - 313)^{0.2}}$ $z = \frac{\text{tg}(0.4) - \sin(7.4) + 45^{0.04}}{\pi^2 * e^2}$
5	<p>Розв'язати математичні вирази:</p> $x = 3 + 4 + 5 + 3 + 5 - 2 - 62 - 36 + 4 * 5 * 3 * 5$ $y = \frac{(11 * 7 + 66 * (-6)) * 9}{(-13)^{2+5}}$ $z = \frac{\sin(0.3) - \cos - 3.4) - 5^{2*4}}{\pi - e}$

Завдання 2

Номер варіанту	Завдання
1	Програма тестування на знання таблиці множення, програма генерує два однорозрядні числа, користувач повинен ввести їх добуток. Якщо користувач ввів правильний результат, то програма повинна про це йому повідомити, якщо ні то вивести повідомлення про помилку та правильний результат.
2	З клавіатури вводяться три цілих числа, визначити з них мінімальне та максимальне
3	Написати програму для перевірки чи рік nYear, що вводиться з клавіатури є високосним. Рік є високосним якщо він ділиться на 4 без остачі, проте якщо в ньому дві останні цифри рівні «00», то щоб бути високосним йому ще додатково необхідно ділитись на 400 без остачі. Наприклад 2000 – високосний, а от 1900 або 1800 – ні.
4	Користувач вводить з клавіатури символ від 'A' до 'F' який відповідає шістнадцятковому числу ('A'-10, 'B'-11 і тд). При цьому символ може бути як у верхньому так і у нижньому регістрах. Програма повинна вивести значення даного числа в десятковому форматі. Наприклад : 'a' ->10, 'F'->15, 'f'->15.
5	Задано чотири числа в діапазоні від 0 до 999 включно. Написати код програми, що буде повертати «TRUE» якщо хоча б одне число є парним, а інше число ділиться на 7 без остачі.
6	Написати програму для переводу однієї валюти в іншу за допомогою оператора switch. Програма повинна здійснювати конвертацію коштів серед трьох валют долар (usd), євро (eur) та гривні (uah). Створити меню для вибору операції користувачем. Курси валют задати в програмі як константи.

Контрольні запитання.

- 1) Опишіть алгоритм отримання випадкового числа?
- 2) Назвіть бібліотеку для роботи з складними математичними операціями на мові Python.

3) Опишіть роботу на менше 5 функції для вирішення складних математичних операцій

4) Охарактеризуйте алгоритми розгалуження, їх види, переваги та особливості використання.

5) Опишіть оператори мови Python для реалізації алгоритмів розгалуження.

Структура звіту.

- 1) Титулка
- 2) Відповіді на теоретичні запитання
- 3) Лістинг реалізованих програм згідно варіанту
- 4) Скрін-шоти роботи реалізованих програм
- 5) Висновки

Лабораторна робота №3. Тема: Циклічні алгоритми та масиви даних

Мета: Отримати теоретичні знання та практичні навички при реалізації програмних додатків обробки одновимірних та багатовимірних масивів

Завдання: Отримати теоретичні та практичні навички при створенні та обробці масивів та стрічок на мові програмування Python

Теоретичні відомості.

У програмуванні цикл For відомий, як цикл із відомою кількістю повторень. Але в Python цей цикл є більш універсальною інструкцією. Це означає, що цикл For використовується в ті моменти, коли кількість ітерацій відома до входу в цикл, на відміну від While циклу, який заснований на умовах.

Конструкція циклу For будується так:

```
for 'змінна' in 'послідовність':  
    'дія'
```

Розглянемо приклад виконання циклу усередині командного інтерпретатора Python.

Напишемо маленький шматок коду, в якому починається цикл for, за ним надається змінна n і в послідовності від 6 до 11 виводяться цифри і за допомогою функції print() повертаємо значення термінал.

```
for n in range(6,11):  
    print(n)
```

У програмуванні індекс чисел починається з 0, тому в результаті ми бачимо 5 різних значень, які знаходяться в діапазоні від 6 до 11.

Використання range() у циклі for

Одним із вбудованих послідовностей є `range()` для управління тим, скільки разів повторюватиметься наш цикл.

У `range()` є три параметри:

`Start` – є початком послідовності (не обов'язковий параметр та значення за умовчанням 0)

`Stop` – це точка зупинки послідовності (обов'язковий параметр)

`Step` – крок послідовності (не обов'язковий параметр і значення за замовчуванням 1).

Розглянемо код із циклом `for`, в якому присвоюємо змінну `n` і значення 15 (параметр `stop`) для `range()`, щоб послідовність була в цьому діапазоні:

```
for n in range(15):  
    print(n)
```

Для зупинки циклу вказали параметр 15 тому в результаті отримали 15 значень від 0 до 14 виключаючи число 15.

Розглянемо код з використанням параметрів `range(start/stop)`, визначимо цикл `for`, надамо змінну `n` і вкажемо значення у вигляді параметрів `start` та `stop`.

```
for n in range(10,15):  
    print(n)
```

Коли ми визначили точку `start` і `stop`, ми отримали результат у діапазоні від 10 до 15, крім числа 15.

Параметр `step` використовується для вказівки кроку між значеннями `start` та `stop`. Розглянемо в наступному коді використання кроків у діапазоні, оголосимо цикл `for` і в ньому змінну `n`, надамо значення параметрам `start/step/stop`:

```
for n in range(25,50,5):  
    print(n)
```

В результаті отримуємо 5 різних значень в діапазоні 25-50 з використанням значення 5 як кроки.

Як параметр `step` можна вказати негативне число, як показано в наступному прикладі. Оголосимо цикл `for` і змінну, параметри `range` привласним значення 1000,900,-10. Щоб ітерація виконувалася у зворотному порядку:

```
for n in range(1000,900,-10):  
    print(n)
```

Ми розглянули програму, де точкою початку 1000 і за допомогою зворотної ітерації отримали значення до 910.

Використання списку в циклі. У циклах можна використовувати список як параметри ітерації. Замість `range()` можна позначити змінну до циклу `for` і використовувати після змінну.

Розглянемо код зі змінним `list`, який приймає кілька слів, далі оголосимо цикл `for` зі змінним `list` і пройдемося за даними цього списку:

```
list = ['one', 'two', 'three', 'four']  
for list in list:  
    print(list)
```

У цьому прикладі ми розглянули виклик даних зі списку, позначеного заздалегідь. Замість перемінно `list`, ми могли використовувати іншу змінну, але результат був однаковим.

Оскільки списки та інші типи даних ітеруються, їх можна комбінувати з функцією `range()`.

Виберемо змінну з прикладу вище, оголосимо цикл `for` із змінною `item`, у параметрах `range()` вкажемо довжину списку `list`. У наступному рядку за допомогою `append` до списку змінної `list` додамо слово `five`:

```
list = ['one', 'two', 'three', 'four', 'five']  
  
for item in range(len(list):  
    list.append('six')  
print(list)
```

У цьому прикладі ми додали 5 слів `six`, що дорівнює кількості слів зі списку `list`.

Також цикл `for` ми можемо використовувати для створення списку різних значень. У наступному прикладі позначимо змінну `numbers`, в якому оголосимо список. Додамо цикл `for` і параметр `25` функції `range()`. У наступному рядку запишемо функцію для ініціалізації нашого порожнього списку зі значенням `n`, який доповнюватиме список:

```
numbers = [ ]  
  
for n in range(25):  
    numbers.append(n)  
  
print(numbers)
```

У прикладі ми розглянули змінну без списку, далі додали цикл, в якому діапазон від `0` до `25` і за допомогою `append` додається значення `+1` до `0` і далі.

У такий же спосіб ми можемо перебирати рядок із символами.

Оголосимо змінну `word` і їй надамо слово `quickly`, за допомогою циклу `for` переберемо слово:

```
word = 'quickly'
```

for letter in word:

```
print(letter)
```

У прикладі вище цикл `for` перебрав слово `quickly` і розбив рядки.

У циклі `for` можна використовувати словник. Як відомо у словнику використовується ключ, перше слово до двокрапки, за ним слідує слово присвоєний до ключа. Щоб правильно викликати значення ключа через двокрапки, розглянемо наступний приклад:

```
list_dict = {'name': 'ServerSpace', 'year': '2022'}
```

for key in list_dict:

```
print(key + ':' + list_dict[key])
```

У наведеному вище прикладі змінна `key` використовувалася позначення ключа, а `list_dict[key]` – позначення значень.

Вкладені цикли `For`. Вкладений цикл є циклом всередині циклу `for` і конструкція вкладеного циклу виглядає так:

```
for 'перша змінна для ітерації' in 'вихідний цикл'
```

```
'дія'
```

```
for 'друга змінна для ітерації' in 'вкладений цикл'
```

```
'дія'
```

Програма спочатку виконує першу ітерацію вихідного циклу, потім переходить у зовнішній цикл і виконує всі ітерації, повертається у вихідний цикл і виконує наступну ітерацію і так до кінця, поки ітерація вихідного циклу не завершиться. Розглянемо в наступному прикладі, оголосимо змінну `num` з цілими типами даних і `let` з символами.

У вихідному циклі оголосимо цикл `for` і змінну `num`, у зовнішньому `let`:

```
num = [10, 15, 25]
```

```
let = ['w', 's', 'x']
```

```
for number in num:  
    print(number)  
    for letter in let:  
        print(letter)
```

Ускладнимо завдання зі списком усередині списку. Спробуємо отримати інформацію окремими рядками:

Оголосимо змінну `list_in_list` і додамо всередину змінної нашого списку;

Додамо вихідний цикл зі змінною `list` та `list_in_list`;

У циклі оголосимо `item` усередину `list`;

Викликаємо `item`.

```
list_in_list = [['one', 'two', 'three'], [25,50,75], [0.25, 0.50, 0.75]]
```

```
for list in list_in_list:  
    for item in list:  
        print(item)
```

Індивідуальні завдання.

Варіанти індивідуального завдання, для кожного варіанта, обираються за формулою

$(\text{№_студента_в_журналі} \% \text{кількість_варіантів_завдання}) + 1$

Завдання 1

Номер варіанту	Завдання
1	За допомогою циклу ввести N цілих чисел, та обчислити їх суму. Число N задається користувачем з клавіатури. Результат вивести на екран

2	За допомогою циклу ввести N цілих чисел, та обчислити суму чисел які вводились з непарним порядковим номером. Число N задається користувачем з клавіатури. Результат вивести на екран
3	За допомогою циклу ввести N цілих чисел, та обчислити суму чисел які вводились з парним порядковим номером. Число N задається користувачем з клавіатури. Результат вивести на екран
4	За допомогою циклу вводити числа, та виводити їх з протилежним значенням. Ввід зупинити якщо було введено N чисел. Число N задається користувачем з клавіатури.
5	За допомогою циклу вводити числа, та виводити значення їх квадратів. Ввід зупинити якщо було введено N чисел. Число N задається користувачем з клавіатури.

Завдання 2

Номер варіанту	Завдання
1	За допомогою циклу вводити цілі числа з клавіатури, зупинити ввід коли введене число буде рівне N . Число N задається користувачем з клавіатури. Кількість введених чисел вивести на екран.
2	За допомогою циклу вводити цілі числа з клавіатури, зупинити ввід коли введене число буде менше за N . Число N задається користувачем з клавіатури. Кількість введених чисел вивести на екран.
3	За допомогою циклу вводити цілі числа з клавіатури, зупинити ввід коли сума введених чисел стане більшою за N . Число N задається користувачем з клавіатури. Кількість введених чисел вивести на екран.
4	За допомогою циклу вводити цілі числа з клавіатури, зупинити ввід коли буде введено N чисел. Число N задається користувачем з клавіатури. Вивести на екран суму додатніх чисел, що були введені з клавіатури.
5	За допомогою циклу вводити цілі числа з клавіатури, зупинити ввід коли буде введено N чисел. Число N задається користувачем з клавіатури. Вивести на екран кількість додатніх та від'ємних чисел, що були введені з клавіатури.

Завдання 3

Номер варіанту	Завдання
1	Задано одномірний масив цілих чисел. Обчислити середнє значення елементів масиву. Результат вивести на екран.
2	Задано одномірний масив символів. Обчислити кількість голосних букв в даному масиві. Результат вивести на екран.
3	Задано одномірний масив цілих чисел. Провести дзеркальне відображення масиву (поміняти місцями перший та останній елементи, другий та передостанній і т.д.). Результат вивести на екран.
4	Задано одномірний масив цілих чисел. Визначити мінімальний елемент масиву. Вивести на екран порядковий номер елемента в масиві та його значення.
5	Задано одномірний масив цілих чисел. Обчислити кількість елементів, що є більшими за N. Число N задається користувачем з клавіатури. Результат вивести на екран.

Завдання 4

Номер варіанту	Завдання
1	Задано стрічку. Обчислити кількість слів та розділових знаків, що знаходять в ній.
2	Задано стрічку. Вивести на екран всі слова, що розпочинаються на голосні літери
3	Задано стрічку. Вивести на екран всі слова, що містять літеру N. Літера N вводиться з клавіатури.
4	Задано стрічку. Вивести на екран всі слова, що є складаються більше чим з N літер. Число N вводиться з клавіатури.
5	Задано стрічку. Вивести на екран всі слова, що містять тільки однакові голосні букви (наприклад, молоко, карат тощо).

Завдання 5

Номер варіанту	Завдання
1	Задано стрічку. Замінити всі голосні букви на літеру N, значення даної літери вводиться з клавіатури. Редаговану стрічку вивести на екран.

2	Задано стрічку. Замінити всі слова довжина яких N на символ “*”. Число N вводиться з клавіатури. Редаговану стрічку вивести на екран.
3	Задано стрічку. Замінити всі слова, що розпочинаються на літеру N, значення даної літери вводиться з клавіатури. Редаговану стрічку вивести на екран.
4	Задано стрічку. Обчислити кількість слів що повторюються. Кількість та самі слова вивести на екран.
5	Задано стрічку. Замінити всі малі літери на великі та навпаки. Редаговану стрічку вивести на екран.

Контрольні запитання.

- 1) Що таке масив та які види масивів існують?
- 2) Основні функції обробки списків.
- 3) Циклічні операції на мові Python.
- 4) Оператори умови на Python.
- 5) Що таке стрічка?
- 6) Опишіть операції на стрічками на мові Python

Структура звіту.

- 6) Титулка
- 7) Відповіді на теоретичні запитання
- 8) Лістинг реалізованих програм згідно варіанту
- 9) Скрін-шоти роботи реалізованих програм
- 10) Висновки

Лабораторна робота №4. Одноплатні комп'ютери Raspberry Pi

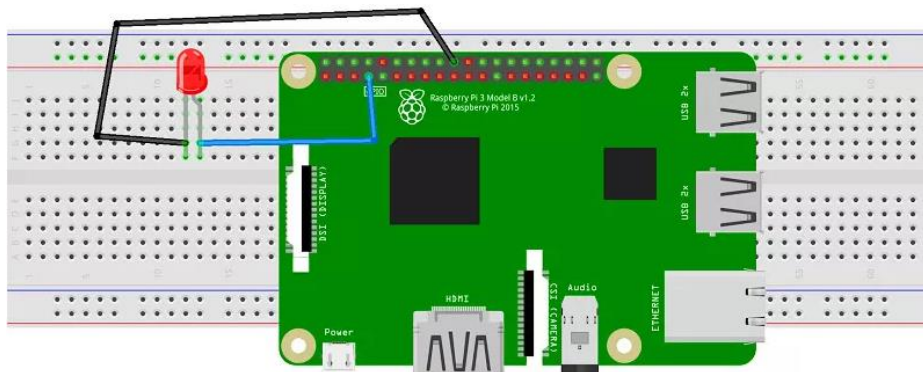
Мета: Отримати теоретичні знання та практичні навички по розробці та реалізації системи на основі одноплатного комп'ютера Raspberry Pi

Завдання: Отримати теоретичні знання та практичні навички при проектуванні програмно-апаратних систем на основі Raspberry Pi за допомогою мови Python

Теоретичні відомості.

Розглянемо реалізацію простого проекту для отримання ефекту миготіння світлодіодом за допомогою Python

По-перше, нам потрібно підключити світлодіод до Raspberry Pi згідно зі схемою нижче:



Тепер все, що вам потрібно зробити – це відкрити редактор Python IDLE (як показано вище), відкрити нове вікно. Далі вводимо наведений нижче код та зберігаємося:

```
import RPi.GPIO як GPIO  
import time  
  
# функція миготіння  
def blink(pin):  
    GPIO.output(pin,GPIO.HIGH)
```

```
time.sleep(1)
GPIO.output(pin,GPIO.LOW)
time.sleep(1)
return

# використовувати номер пина Raspberry Pi
GPIO.setmode(GPIO.BOARD)

# Налаштування виходу GPIO
GPIO.setup(11, GPIO.OUT)

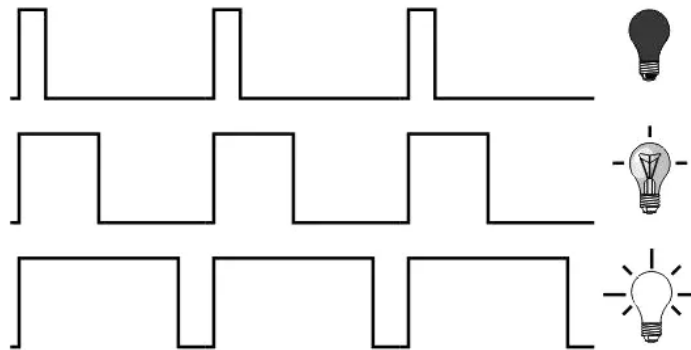
# блимати GPIO17 50 разів
for i in range(0,50):
    blink(11)
GPIO.cleanup()
```

Збережіть файл та запустіть його, натиснувши F5. Ви повинні побачити, як ваш світлодіод блимає. Це простий приклад першого апаратного проекту на Raspberry Pi. Тепер можна піти далі і скористатися Широтно-Імпульсною Модуляцією (ШИМ), щоб світлодіод міг загасати.

ШИМ або широтно-імпульсна модуляція (англ. – PWM або Pulse Width Modulation) – це тип цифрового сигналу. Цифровий сигнал може мати лише два можливі стани: УВІМК або ВИМК, 0 або 1, або, у разі цього проекту, 0 або 5 вольт. Ось чому для того, щоб світлодіод блимав, не був потрібен ШИМ, тому що світлодіод просто вмикався і вимикався. У сигналах ШИМ ми можемо мати обидва ці стани протягом певного періоду часу.

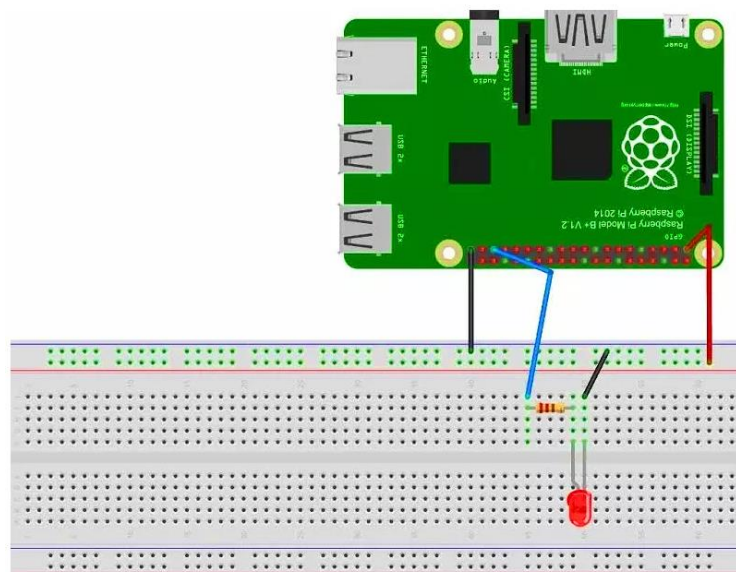
Припустимо, ви хочете керувати яскравістю світлодіода. Можливий підхід у тому, щоб увімкнути світлодіод на невеликий період часу, а потім знову вимкнути його на невеликий проміжок часу.

Таким чином, коли це включення та вимкнення відбувається на дуже високій швидкості, це дає ефект загасання світлодіода. Подивіться на малюнок нижче.



Чим ширша прямокутна хвиля, тим яскравіше світиться світлодіод.

Схема з'єднання



Підключіть Raspberry Pi та світлодіод, як показано на схемі нижче. Світлодіод підключено до GPIO19.

Після підключення схеми відкрийте Python 3 та натисніть CTRL+N, щоб відкрити нове вікно. Скопіюйте наведений нижче код у вікно:

```

import RPi.GPIO as IO
import time

IO.setwarnings(False)
IO.setmode (IO.BCM)

# ініціалізація GPIO19 як вихід
IO.setup(19,IO.OUT)

# Частота 100Hz
p = IO.PWM(19,100)
p.start(0)

while 1:
    for x in range (50):
        p.ChangeDutyCycle(x)
        time.sleep(0.1)

    for x in range (50):
        p.ChangeDutyCycle(50-x)
        time.sleep(0.1)

```

Коли ви запускаєте програму, ваш світлодіод повинен змінювати стан від яскравого до темного.

Індивідуальні завдання.

Варіанти індивідуального завдання, для кожного варіанта, обираються за формулою

(№_студента_в_журналі%_кількість_варіантів_завдання)+1

Завдання 1

Номер варіанту	Завдання
1	Пристрій для детекції руху який при виявленні повинен включати світловий сигнал
2	Пристрій для аналізу рівня яскравості який при пониженні рівня за заданий поріг, включається звуковий сигнал
3	Пристрій для аналізу рівня температури який при пониженні рівня за заданий поріг, включається звуковий сигнал
4	Пристрій для подання звукового сигналу за заданий період часу (пристрій -будильник)
5	Пристрій для моніторингу рівня води в ємності який при пониженні/перевищенні рівня за заданий поріг, включається звуковий сигнал

Контрольні запитання.

- 1) Які особливості при роботі з одноплатним комп'ютером Raspberry Pi?
- 2) Назвіть та опишіть основні технічні особливості програматорів для одноплатних комп'ютерів Raspberry Pi.
- 3) Опишіть основні етапи створення програмно-апаратних систем на основі Raspberry Pi
- 4) Опишіть реалізований вами проект.

Структура звіту.

- 11) Титулка
- 12) Відповіді на теоретичні запитання
- 13) Лістинг реалізованих програм згідно варіанту
- 14) Скрін-шоти роботи реалізованих програм
- 15) Висновки

РЕКОМЕНДОВАНИ ДЖЕРЕЛА ИНФОРМАЦІЇ

1. Al Sweigart *Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code*, No Starch Press (December 16, 2020); eBook (Read Online). 2020. 384.
2. Tomas Beuzen, Tiffany Timbers *Python Packages*. Chapman and Hall/CRC; 1st edition (April 21, 2022); eBook (Open Source Book). 2022. 222.
3. Harry Percival, Bob Gregory Francois Chollet *Deep Learning with Python*, Second Edition. Manning; 2nd edition (December 21, 2021). 2021. 504.
4. *Architecture Patterns with Python: Enabling Test-Driven Development, Domain-Driven Design, and Event-Driven Microservices*. O'Reilly Media; 1st edition (2020); eBook (Creative Commons Licensed). 2020. 304.
5. Leonardo Giordani *Clean Architectures in Python: A Practical Approach to Better Software Design*. The Digital Cat Books/Leanpub (2022). 2022. 422.
6. Yasoob Khalid *Practical Python Projects*. Self-Publishing, 2022. 2022.
7. Lee Vaughan *Impractical Python Projects: Playful Programming Activities to Make You Smarter* Illustrated Edition. No Starch Press; Illustrated edition (November 27, 2018). 2018. 424.
8. . Al Sweigart *The Big Book of Small Python Projects: 81 Easy Practice Programs* No Starch Press (June 25, 2021); eBook (Read Online). 2021. 432.
9. Gareth Halfacre *The Official Raspberry Pi Beginner's Guide* Raspberry Pi Press (2019); eBook (4th Edition, 2020, Creative Commons Licensed). 2019. 241.
10. Rob Zwetsloot *The Official Raspberry Pi Handbook* Raspberry Pi Press (2022); eBook (Creative Commons Licensed). 2022. 241.