

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»  
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Комп'ютерний практикум №1  
з курсу алгоритми кодування двійкових  
даних

**РЕАЛІЗАЦІЯ АЛГОРИТМУ  
BASE64**

Виконав студент  
групи ФІ-42мн  
Бєш Радомир Андрійович

Київ — 2026

## Зміст

<b>1</b>	<b>Мета</b>	<b>3</b>
<b>2</b>	<b>Постановка задачі</b>	<b>3</b>
<b>3</b>	<b>Хід роботи</b>	<b>3</b>
3.1	Вимоги для кодера . . . . .	3
3.2	Результати кодування . . . . .	4
<b>4</b>	<b>Вимоги для декодера</b>	<b>5</b>
4.1	Результати декодування . . . . .	7
<b>5</b>	<b>Висновки</b>	<b>7</b>

## 1 Мета

Опанувати поняття коду, дослідити різні види перетворень даних, представлених двійковими словами (байтами).

## 2 Постановка задачі

Реалізувати програмно алгоритм кодування даних за Base64.

**Результати дослідження:**

[https://github.com/Radomir21/Encoding-Algorithms/tree/main/lab\\_1](https://github.com/Radomir21/Encoding-Algorithms/tree/main/lab_1).

## 3 Хід роботи

### 3.1 Вимоги для кодера

Base64-кодер повинен бути реалізований як окрема програма або як модуль в межах більш функціональних програм, який викликається спеціальною командою (з інтерфейсу програми, з меню чи з командного рядку консолі).

- Кодер повинен приймати на вхід ім'я файла для кодування та (опціонально) ім'я файла для зберігання закодованого результату.

```

> _pycache_
└─ Encoding Algo\Encoding-Algorithms
    └─ lab_1
        > _pycache_
            ⚡ decode_base64.py
            ⚡ decoded_file.txt
            ⚡ encode_base64.py
            ⚡ file_to_encode.txt
            ⚡ input.base64
            ⚡ output.base64
            ⚡ .gitignore
            ⓘ README.md
PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab_1> python encode_base64.py
Введі ім'я вхідного файла: file_to_encode.txt
Введі ім'я вихідного файла (або Enter): decoded_file.txt
Файл успішно закодовано -> decoded_file.txt
PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab_1>

```

- Якщо користувач не надав ім'я файла для зберігання закодованого результату, кодер повинен створити такий файл сам, додавши до імені оригінального файла розширення ".base64".

```

└─ VS CODE PROJECTS
    > _pycache_
    └─ Encoding Algo\Encoding-Algorithms
        └─ lab_1
            > _pycache_
                ⚡ decode_base64.py
                ⚡ decoded_file.txt
                ⚡ encode_base64.py
                ⚡ file_to_encode.txt
                ⚡ file_to_encode.txt.base64
                ⚡ input.base64
                ⚡ output.base64
                ⚡ .gitignore
                ⓘ README.md
PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab_1> python encode_base64.py
Введі ім'я вхідного файла: file_to_encode.txt
Введі ім'я вихідного файла (або Enter):
Файл успішно закодовано -> file_to_encode.txt.base64
PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab_1>

```

- Вхідний файл повинен трактуватись як файл двійкових даних (потік байтів).

```

51
52     with open(input_file, "rb") as file:
53         data = file.read()

```

4. Результат кодування повинен записуватись у файл виходу як у текстовий файл, рядками по 76 символів (окрім, можливо, короткого останнього рядку). Для цього була написана функція `def split_lines_76()`.

```

12  def split_lines_76(encoded: str) -> str:
13
14      lines = []
15      for i in range(0, len(encoded), 76):
16          line = encoded[i:i+76]
17          if len(line) == 0:
18              continue
19          if len(line) > 76:
20              raise ValueError("Довжина рядка перевищує 76 символів")
21          lines.append(line)
22      return lines

```

Після кодування результат записуємо у вихідний файл:

```

63      for line in lines:
64          file.write(line + "\n")

```

5. Кодер може вставляти у файл виходу довільні коментарі (метадані). Кожен рядок повинен починатись з символу «-» та не перевищувати 76 символів.

```

3  def ensure_valid_comment(comment: str) -> str:
4
5      if not comment.startswith("-"):
6          raise ValueError("Коментар повинен починатися зі знака '-'")
7
8      if len(comment) > 76:
9          raise ValueError("Коментар занадто довгий (макс 76 символів)")
10

```

Далі слідує перевірка на валідність коментаря:

```

58      with open(output_file, "w", encoding="ASCII") as file:
59          if comment is not None:
60              ensure_valid_comment(comment)
61              file.write(comment + "\n")

```

## 3.2 Результати кодування

Була закодована послідовність символів:

```
1  AAAAAAAAAAAAAAAfsfsfsdfsadadadDDDDDD3453524242424242424
```

Вихідний файл містить наступну закодовану послідовність з коментарем:

```

1  -my comments
2  QUFBQUFBQUFBQUFBQUFBZnNm2ZZGZZmRzYWRhZGFkREREREQzNDUzNTI0MjQyNDI0MjQy
3  NA==
4

```

## 4 Вимоги для декодера

Base64-декодер повинен бути реалізований як окрема програма або як модуль у більш функціональних програмах, який викликається спеціальною командою.

- Декодер повинен приймати на вхід ім'я закодованого файлу та ім'я файла для зберігання декодованого результату.

```

> _pycache__
> Encoding_Algo\Encoding-Algorithms
  > lab_1
    > _pycache__
      decode_base64.py
      decoded_file.txt
      encode_base64.py
      file_to_encode.txt
      file_to_encode.txt.base64
      input.base64
      output.base64
      sers/XD182UX\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/XD182UX/OneDrive - EY/Documents/VS Code projects/Encoding_Algo-Encoding-Algorithms/lab_1/decode_base64.py"
      Введі ім'я закодованого файла: input.base64
      Запропоноване ім'я вихідного файла: output.base64
      Використати це ім'я? (y/n): y
      Файл успішно декодовано -> output.base64
      PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding_Algo-Encoding-Algorithms\lab_1>
  
```

- Декодер може запропонувати користувачу варіант імені для декодованого файлу:

```

VS CODE PROJECTS
> _pycache__
> Encoding_Algo\Encoding-Algorithms
  > lab_1
    > _pycache__
      decode_base64.py
      decoded_file.txt
      encode_base64.py
      file_to_encode.txt
      file_to_encode.txt.base64
      input.base64
      newoutput.base64
      output.base64
      PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding_Algo-Encoding-Algorithms\lab_1> python decode_base64.py
      Введі ім'я закодованого файла: input.base64
      Запропоноване ім'я вихідного файла: output.base64
      Використати це ім'я? (y/n): n
      Введі ім'я вихідного файла: newoutput.base64
      Файл успішно декодовано -> newoutput.base64
      PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding_Algo-Encoding-Algorithms\lab_1>
  
```

- Файл виходу повинен трактуватись як файл двійкових даних (потік байтів).

109	with open(output_file, "wb") as f:
110	f.write(decoded)

- Вхідний файл повинен трактуватись як текстовий файл з результатами Base64-кодування та рядками-коментарями.

1	-my comments
2	QUFBQUFBQUFBQUFBQUFBQUFBZnNmC2ZZGZzZmRzYWRhZGFkREREREQzNDUzNTI0MjQyNDI0MjQy
3	NA==
4	

- Рядки, що починаються з «-», повинні ігноруватись декодером.

74	if line.lstrip().startswith("-"):
75	continue
76	

Інші рядки, які містять «-» не на першому місці, трактуються як помилка «Некоректний вхідний символ».

90	if "-" in line:
91	pos = line.index("-") + 1
92	print_invalid_char(idx, pos, "-")
93	

```

6     def print_invalid_char(line, pos, ch):
7         print(f"Рядок {line}, символ {pos:02d}: Некоректний вхідний символ ('{ch}')")
8         sys.exit(0)

```

6. Усі рядки (крім останнього) повинні мати довжину 76 символів.

```

14     def print_length_error(line, length):
15         print(f"Рядок {line}: Некоректна довжина рядка ({length})")
16         sys.exit(0)

```

7. Якщо рядок має символ, який не належить алфавіту, декодер повертає помилку «*Некоректний вхідний символ*».
8. Символи « = » означають завершення закодованого повідомлення. Неправильне використання падінгу викликає помилку.

```

94     if idx != last_real_line_index and len(line) != 76:
95         print_length_error(idx, len(line))

```

9. Якщо після ідентифікованого рядка ще є дані, що не є коментарями, декодер повертає попередження «*Наявні дані після кінця повідомлення*». і не декодувати такі дані. При цьому результат декодування правильної частини вхідного файлу повинен бути записаний у файл виходу.

## 4.1 Результати декодування

Декодуємо наступний код:

```

Encoding Algo > Encoding-Algorithms > lab_1 > input.base64
1 -smthing comments
2 TWFuIGzIGRpC3Rpimd1aXNoZwQsIG5vdCBvbmx5IGJ5IGHpcyByZWfzb24sIGJ1dCBiesB0aglz
3 IHNpbmd1bgFyIHBhc3Npb24gZnJvbsBvdgh1cIBhbmltvWxzLCB3aG1jaCBpcyBhGx1c3Qgb2Yg
4 dGhlIG1pbmQsIHROYXQgYnkgySBwZXJZZXlcmFuy2Ugb2YgZGVsaWdodCBpb1B0aGUGy29udglu
5 dwVkiGfUzC8bmRlzmf0awdhYmxl1Gd1mVyxRpb24gb2Yga25vd2x1Zgd1LlCbleGNlZWRzIHRo
6 ZSBzaG9ydcB2Zwh1bwvuy2Ugb2YgV55IGNhcm5hbCBwbGvhc3VyzSA=
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python - lab\_1 + ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

- PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab\_1> **python decode\_base64.py**  
Введі ім'я закодованого файла: input.base64  
Запропоноване ім'я вихідного файла: output.base64  
Використати це ім'я? (y/n): y  
файл успішно декодовано -> output.base64
- PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab\_1>

Результат декодування:

```

Encoding Algo > Encoding-Algorithms > lab_1 > output.base64
1 Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python - lab\_1 + ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

- PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab\_1> **python decode\_base64.py**  
Введі ім'я закодованого файла: input.base64  
Запропоноване ім'я вихідного файла: output.base64  
Використати це ім'я? (y/n): y  
файл успішно декодовано -> output.base64
- PS C:\Users\XD182UX\OneDrive - EY\Documents\VS Code projects\Encoding Algo\Encoding-Algorithms\lab\_1>

## 5 Висновки

У даній лабораторній роботі було реалізовано алгоритм Base64-кодування та декодування мовою Python. Вхідні дані обробляються як двійковий потік байтів, а результати коректно зберігаються у вихідні файли.

Під час реалізації декодера було виконано перевірку довжини рядків, допустимих символів, правильності використання паддінгу та обробку рядків-коментарів. Також передбачено виведення попередження у разі наявності зайвих даних після завершення повідомлення.

Результати тестування підтвердили правильність роботи програми та відповідність реалізації вимогам методичних вказівок. Труднощі виникли тільки з таймінгами виконання даної роботи.