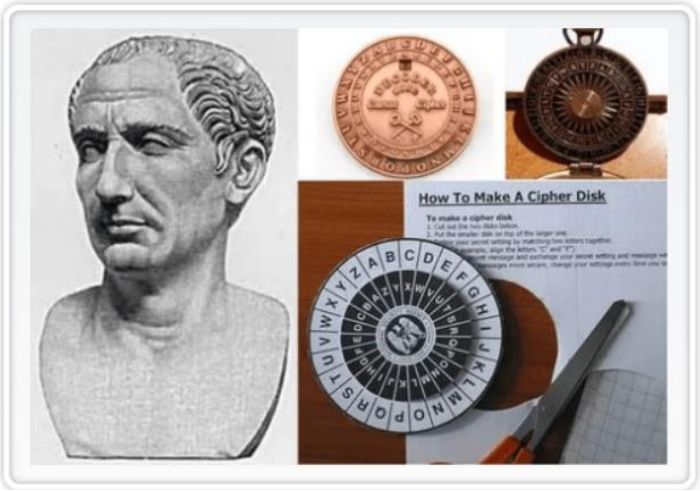
Урок 40. Шифр Цезаря

Вступна частина	**Вітаємо в IT-книзі!** На минулих уроках ми виконували різноманітні дії зі списками, що містять числа. Сьогодні ви будете працювати з текстовими даними. Перетворившись на справжніх секретних агентів, ви створите програму для шифрування повідомлень. Сьогодні на уроці вам буде запропоновано виконати 13 завдань, кожне з яких оцінюється 1 балом, та 1 завдання без оцінки. Таким чином, за цей урок можна отримати більше 12 балів. Однак якщо підсумковий результат вас не задовольнить, то наприкінці уроку ви зможете покращити свій результат, виконавши додаткове завдання. Деякі завдання потрібно виконати в середовищі Python. Їх позначено олівцем.

Вивчення нового матеріалу

Слайд № 1	Сьогодні ми створимо програму для шифрування текстових повідомлень. Зараз одним із найпростіших вважається **шифр Цезаря**, однак за часів Давнього Риму цей шифр не міг розшифрувати ніхто. 

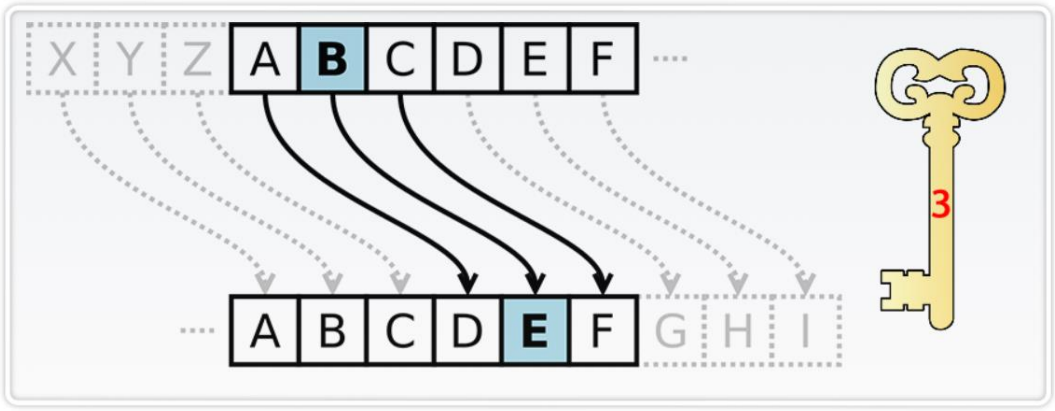
Слайд № 2

Зашифрувати повідомлення - це все одно, як замкнути його в скрині: прочитати його зможе лише той, хто має ключ.

Ключем називають невеликий обсяг інформації, за допомогою якого шифрують та розшифровують повідомлення.

Слайд № 3

Принцип шифрування за допомогою **шифру Цезаря**: кожну літеру тексту замінити на ту, яка в алфавіті йде від неї на k літер далі. **k — це ключ**.



Слайд № 4

Ми запрограмуємо такий варіант шифру Цезаря:
під час шифрування кожна літера замінюється іншою, що розташована від неї на **k** символів **Unicode**.

Величина **k** — це ключ.

літера	а	б	в	г	д	е	ж	з	и	й
код	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081
літера	к	л	м	н	о	п	р	с	т	у
код	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091
літера	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
код	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101
літера	ю	я	è	ё	ђ	ѓ	є	ѕ	і	ї
код	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111
літера	ј	љ	њ	ћ	ќ	ѝ	ў	џ	Ѡ	ѡ
код	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121
літера	Ъ	ъ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ
код	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131
літера	Ѭ	ѭ	Ѯ	ѯ	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ
код	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141

Слайд № 5

Наприклад, зашифруємо у такий спосіб слово **"шифрування"**.

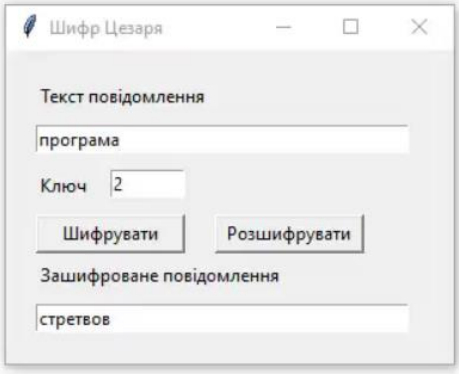
Ключ шифру **k=2**.

літера	а	б	в	г	д	е	ж	з	и	й
шифр	в	г	д	е	ж	з	и	й	к	л
літера	к	л	м	н	о	п	р	с	т	у
шифр	м	н	о	п	р	с	т	у	ф	х
літера	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
шифр	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
літера	ю	я	є	ё	ђ	ѓ	ѕ	і	ї	
шифр	є	ё	ђ	ѓ	ѕ	і	ї	ј	љ	
літера	ј	љ	њ	ћ	ќ	ѝ	ў	џ	Ѡ	ѡ
шифр	њ	ћ	ќ	ѝ	ў	џ	Ѡ	ѡ	Ѣ	ѣ
літера	ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ
шифр	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ
літера	Ѭ	ѭ	Ѯ	ѯ	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ
шифр	Ѯ	ѯ	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	←	на 2

Відповідь: **ъкцтхдвппё**

Слайд № 6

Розглянемо, як має працювати наша програма-шифрувальник.

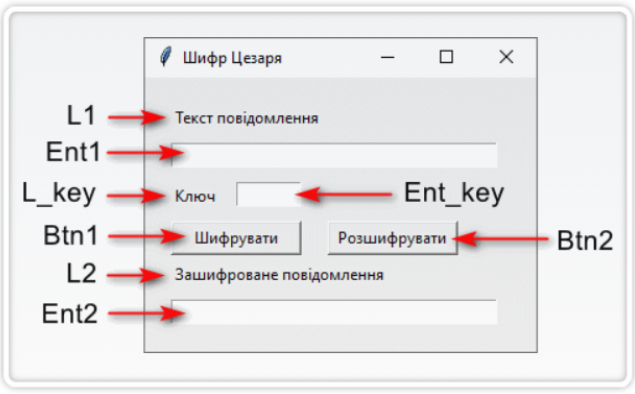


Вправа

Вправа № 1

Вправа 1. Створіть інтерфейс вікна програми відповідно до зразка.

Надайте елементам керування імена, що вказані у виносках.



Вивчення нового матеріалу

Слайд № 7

Перш ніж програмувати обробник події натискання кнопки **Шифрувати**, розглянемо принцип зберігання текстових даних.

Ось так, наприклад, у пам'яті комп'ютера буде записано рядок, який містить змінна **a**:

```
a="моя програма"
```

i	0	1	2	3	4	5	6	7	8	9	10	11
a	м	о	я		п	р	о	г	р	а	м	а

тут **i** – це порядковий номер символу рядка **a**. Символи нумеруються з нуля.

```
>>> a="моя програма"
>>> s=a[4]
>>> print(s)
п
```

Отже, текстовий рядок — це список символів. Щоб отримати символ рядка, необхідно вказати ім'я змінної, що містить рядок, а в квадратних дужках — індекс символу.

Слайд № 8

Для перебору усіх симоволів рядка використовують цикл, що має такий вигляд:

```
for лічильник in ім'я рядка:
    тіло циклу
```

Так, наприклад, буде виглядати цикл, який шукає, скільки разів у рядку **x** трапляється літера "а":

```
for i in x:
    if i=="a":
        k=k+1
```

лічильник — for i; ім'я рядка — in x; тіло циклу — k=k+1

Увага!

У наведеному циклі для доступу до символу рядка використовується лічильник **i**, а не вираз **s[i]**

Слайд № 9

Як ви вже знаєте з попередніх тем, кожен символ має числовий код. Коди ставляться у відповідність символам за допомогою кодової таблиці.

Ось, наприклад, фрагмент кодової таблиці Unicode.

літера	а	б	в	г	д	е	ж	з	и	й
код	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081
літера	к	л	м	н	о	п	р	с	т	у
код	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091
літера	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
код	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101
літера	ю	я	ѐ	ё	ђ	ѓ	є	ѕ	і	ї
код	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111
літера	ј	љ	њ	ћ	ќ	й	ў	џ	Ѡ	ѡ
код	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121
літера	Ъ	ъ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ
код	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131
літера	Ѭ	ѭ	Ѯ	ѯ	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ
код	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141

Слайд № 10

Отже, ми визначили, що під час шифрування рядка s1 для кожного його символу необхідно:

1) визначити код символу та додати до нього ключ;

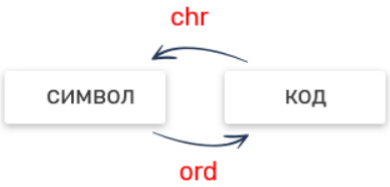
2) одержати значення зашифрованого символу за його кодом та дописати його до шифрованого тексту.

Для визначення **коду символу** використовується функція **ord(x)**, де **x** – символ.

Для визначення **символу** Unicode за його кодом – **chr(y)**, де **y** – код символу.

Наприклад,

```
ord("a")=97
chr(97)="a"
```



Вправа

Вправа № 2

Вправа 2. Додайте до програми код обробника події натискання кнопки Шифрувати та перевірте правильність його виконання.

Алгорим:

1. Присвоїти змінній **key** значення ключа, що міститься в полі **Ent_key**.
2. Присвоїти змінній **s1** текст, який містить поле **Ent1**.
3. Присвоїти змінній **s2** порожній рядок.
4. Записати коди шифрування кожного символу і присвоєння зашифрованого тексту змінній **s2**.
5. Вивести рядок **s2** в полі **Ent2**.

Підказка до вправи №2

Щоб створити обробник події натискання кнопки,

1) Створіть на початку програми функцію. Ось її заголовок:
   `def Btn1_Click():`
   У тілі цієї функції і повинен розміщуватися наведений нижче код.
2) Додайте параметр прив'язки цієї функції до конструктора кнопки:
   `command=Btn1_click`.

```
s1=Ent1.get()
```

```
s2=""
```

```
for i in s1:
    kod=ord(i)+key
    s2=s2+chr(kod)
```

```
Ent2.insert(0,s2)
```

```
key=int(Ent_key.get())
```

Ось фрагменти коду, з яких потрібно сформувати алгоритм.