МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІКТА

Кафедра ЗІ

Звіт

до лабораторної роботи №4

з курсу: «Програмування скриптовими мовами» на тему: «Написання програм з умовними виразами та циклами»

Виконав:

студент групи КБ-305

Семенчук А.А.

Прийняв:

к.т.н., доцент

Совин Я. Р.

Львів 2024

Мета роботи – ознайомитись з вбудованими рядковими типами Python та операторами і функціями для роботи з ними.

Завдання

1. Написати програму табулювання функції (див. табл. 3), що вибирається залежно від значення аргументу, на проміжку [a, b] з кроком табуляції h. При табулюванні має виводитися аргумент x, значення функції у з точністю 4 знаки після коми. Ширина полів аргументу і значення функції має бути фіксована і вирівняна.

2. Написати програму табулювання функції, представленої рядом (див. табл. 4), на інтервалі [a, b] з кроком табуляції h та абсолютною похибкою d. Оцінку похибки здійснювати за значенням модуля чергового члена ряду. При табулюванні має виводитися аргумент x, значення функції у та абсолютна похибка d з точністю 5 знаків після коми. Ширина полів аргументу, значення функції і похибки має бути фіксована і вирівняна.

3. Написати програму валідації введеного паролю. Програма не повинна використовувати регулярні вирази, списки, множини, словники, функції, класи чи сторонні бібліотеки окрім colorama. Користувач повинен ввести пароль, програма має перевірити наявність у ньому лише заданих типів символів у вказаних пропорціях і з дотримання додаткових правил згідно варіанту у табл. 5 і вивести інформацію про результати перевірки у форматі як показано на рис. 1.

$\frac{1+\cos(x)}{1-\sin(x^2)}$ $\operatorname{arctg}(\ln(x+\sin(x)))$ $\frac{1}{x^2}+\frac{1}{x^3}+\frac{1}{x^4}$	$x < 4$ $4 \le x < 5$ $x \ge 5$	0.2	[3, 6]
$1+\sum_{n=1}^{\infty}\frac{x}{(3n-1)^3}\sin(n+2)$		[0.1, 0.6]	0.05	0.001

27.	10	low_char + num_char + spc_char + upp_char	1, 13, 9, 20, 11, 22, 5, 18, 27, 34

Виконання лабораторної роботи

1.

```python
import math

# Функція для обчислення значення залежно від x
def calculate_function(x):
    if x < 4:
        return (1 + math.cos(x)) / (1 - math.sin(x ** 2))
    elif 4 <= x < 5:
        return math.atan(math.log(x + math.sin(x)))
    else:
        return 1 / (x ** 2) + 1 / (x ** 3) + 1 / (x ** 4)
```

```python
# Параметри
a, b = 3, 6
h = 0.2

# Табулювання
x = a
print(f"{'x':<10}{'y':<15}")
while x <= b:
    y = calculate_function(x)
    print(f"{x:<10.1f}{y:<15.4f}")
    x += h
```

```
@Velovo123 →/workspaces/python-labs/lab4 (main) $ python tabulate_function.py
x         y
3.0       0.0170
3.2       0.0010
3.4       0.0180
3.6       0.1675
3.8       4.5937
4.0       0.8663
4.2       0.8771
4.4       0.8913
4.6       0.9086
4.8       0.9283
5.0       0.0496
5.2       0.0455
5.4       0.0418
5.6       0.0386
5.8       0.0357
```

2.

```python
import math

# Функція для обчислення значення ряду з похибкою
def calculate_series(x, d):
    n = 1
    term = x / ((3 * n - 1) ** 3 * math.sin(n + 2))
    total_sum = 1 + term
    while abs(term) >= d:
        n += 1
        term = x / ((3 * n - 1) ** 3 * math.sin(n + 2))
        total_sum += term
    return total_sum

# Параметри
a, b = 0.1, 0.6  # Інтервал для x
h = 0.05         # Крок табуляції
d = 0.001        # Допустима похибка

# Табулювання
x = a
print(f"{'x':<10}{'y':<15}{'error':<15}")
```

```python
while x <= b:
    y = calculate_series(x, d)
    error = d
    print(f"{x:<10.2f}{y:<15.5f}{error:<15.5f}")
    x += h
```

```
@Velovo123 →/workspaces/python-labs/lab4 (main) $ python tabulate_series.py
x         y               error
0.10      1.08732         0.00100
0.15      1.13097         0.00100
0.20      1.17463         0.00100
0.25      1.21829         0.00100
0.30      1.26195         0.00100
0.35      1.30561         0.00100
0.40      1.34927         0.00100
0.45      1.39292         0.00100
0.50      1.43551         0.00100
0.55      1.47907         0.00100
0.60      1.52262         0.00100
```

3.

```python
from colorama import Fore, Style

# Введення паролю
password = input("Введіть пароль довжиною не менше 10 символів: ")

# Вимоги до символів
upp_char = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
low_char = "abcdefghijklmnopqrstuvwxyz"
num_char = "0123456789"
spc_char = "!@#$%^&*_-"

# Перевірка довжини пароля
length_check = len(password) >= 10

# Перевірка дозволених символів
allowed_chars_check = all(c in (upp_char + low_char + num_char + spc_char) for c in
password)

# Підрахунок кількості кожного типу символів
low_count = sum(1 for c in password if c in low_char)
upp_count = sum(1 for c in password if c in upp_char)
num_count = sum(1 for c in password if c in num_char)
spc_count = sum(1 for c in password if c in spc_char)

# Валідація кожного правила
rules = {
    "Довжина не менше 10 символів": length_check,
    "Пароль містить лише допустимі символи": allowed_chars_check,
    "Не менше 2 маленьких латинських літер": low_count >= 2,
    "Не більше 4 маленьких латинських літер": low_count <= 4,
    "Не менше 2 цифр": num_count >= 2,
    "Не більше 4 цифр": num_count <= 4,
```

```
    "Не менше 2 спеціальних символів": spc_count >= 2,
    "Не більше 3 спеціальних символів": spc_count <= 3,
    "Не менше 2 великих латинських літер": upp_count >= 2,
    "Не більше 5 великих латинських літер": upp_count <= 5,
    "Не більше 3 однакових великих латинських літер підряд": not any(password[i] ==
password[i+1] == password[i+2] for i in range(len(password) - 2) if password[i] in
upp_char),
    "Не більше 3 однакових спеціальних символів підряд": not any(password[i] ==
password[i+1] == password[i+2] for i in range(len(password) - 2) if password[i] in
spc_char)
}

# Вивід результатів перевірки для кожного правила
print("\nВимоги до паролю:")
for rule, passed in rules.items():
    color = Fore.GREEN if passed else Fore.RED
    status = "OK!" if passed else "FAIL!"
    print(f"{rule} - {color}{status}{Style.RESET_ALL}")

# Перевірка загальної валідності пароля
is_valid = all(rules.values())
print(Fore.GREEN + "Пароль валідний!" + Style.RESET_ALL if is_valid else Fore.RED +
"Пароль не валідний!" + Style.RESET_ALL)
```

```
@Velovo123 ➜/workspaces/python-labs/lab4 (main) $ python validate_password.py
Введіть пароль довжиною не менше 10 символів: ab3!cd2@zx

Вимоги до паролю:
Довжина не менше 10 символів - OK!
Пароль містить лише допустимі символи - OK!
Не менше 2 маленьких латинських літер - OK!
Не більше 4 маленьких латинських літер - FAIL!
Не менше 2 цифр - OK!
Не більше 4 цифр - OK!
Не менше 2 спеціальних символів - OK!
Не більше 3 спеціальних символів - OK!
Не менше 2 великих латинських літер - FAIL!
Не більше 5 великих латинських літер - OK!
Не більше 3 однакових великих латинських літер підряд - OK!
Не більше 3 однакових спеціальних символів підряд - OK!
Пароль не валідний!
@Velovo123 ➜/workspaces/python-labs/lab4 (main) $ python vaildate_password.py
Введіть пароль довжиною не менше 10 символів: aB3!bD!!@X

Вимоги до паролю:
Довжина не менше 10 символів - OK!
Пароль містить лише допустимі символи - OK!
Не менше 2 маленьких латинських літер - OK!
Не більше 4 маленьких латинських літер - OK!
Не менше 2 цифр - FAIL!
Не більше 4 цифр - OK!
Не менше 2 спеціальних символів - OK!
Не більше 3 спеціальних символів - FAIL!
Не менше 2 великих латинських літер - OK!
Не більше 5 великих латинських літер - OK!
Не більше 3 однакових великих латинських літер підряд - OK!
Не більше 3 однакових спеціальних символів підряд - OK!
Пароль не валідний!
@Velovo123 ➜/workspaces/python-labs/lab4 (main) $ python vaildate_password.py
Введіть пароль довжиною не менше 10 символів: aB3!bD2@Xz

Вимоги до паролю:
Довжина не менше 10 символів - OK!
Пароль містить лише допустимі символи - OK!
Не менше 2 маленьких латинських літер - OK!
Не більше 4 маленьких латинських літер - OK!
Не менше 2 цифр - OK!
Не більше 4 цифр - OK!
Не менше 2 спеціальних символів - OK!
Не більше 3 спеціальних символів - OK!
Не менше 2 великих латинських літер - OK!
Не більше 5 великих латинських літер - OK!
Не більше 3 однакових великих латинських літер підряд - OK!
Не більше 3 однакових спеціальних символів підряд - OK!
Пароль валідний!
@Velovo123 ➜/workspaces/python-labs/lab4 (main) $
```

Висновок: У ході виконання роботи ми ознайомилися з умовними операторами та циклічними конструкціями в мові Python. Ми навчилися використовувати оператори `if`, `elif` та `else` для реалізації розгалужень у коді, що дозволяє приймати рішення залежно від виконання певних умов. Також ми вивчили цикли `for` і `while`, які дозволяють повторювати виконання певного блоку коду. Завдяки цьому ми змогли розв'язувати задачі, що потребують багаторазового виконання однотипних операцій, та оптимізувати процес перевірки умов, що значно підвищило ефективність написання коду.