

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №5
" ФУНКЦІЇ "

Виконав студент гр. 1БС -206

_____ Дмитришин Є.Т.

Лабораторну роботу захищено

з оцінкою _____

Перевірив

ас. каф. ЗІ _____

Карась О.В.

_____ 2022 р.

Вінниця 2022 р.

Завдання №1

1. Створити функцію обчислення n-го члена геометричної прогресії, коли задано її перший член та знаменник.

Лістинг програми

```
def geom_progression():
    b1 = float(input('Введіть перший член геометричної прогресії: '))
    q = float(input('Введіть знаменник геометричної прогресії: '))
    n = float(input('Введіть номер члена, який потрібно обчислити: '))
    print(b1 * pow(q, n - 1))

geom_progression()
```

Результати роботи програми

```
Введіть перший член геометричної прогресії: 2
Введіть знаменник геометричної прогресії: 4
Введіть номер члена, який потрібно обчислити: 5
512.0
```

Завдання №2

2. Напишіть функцію `is_password_good(password)`, яка приймає як аргумент рядкове значення пароля `password` і повертає значення `True` якщо пароль є надійним і `False` інакше.

Пароль є надійним якщо:

- його довжина щонайменше 8 символів;
- він містить як мінімум одну заголовну літеру (верхній регістр);
- він містить як мінімум одну малу літеру (нижній регістр);
- він містить хоч би одну цифру.

Лістинг програми

```
def is_password_good(password):
    result = True
    if len(password) < 8:
        result = False
    if password.lower() == password:
        result = False
    if password.upper() == password:
        result = False
    if password.isalpha():
        result = False
    return result

txt = input('Введіть пароль: ')
print(is_password_good(txt))
```

Результати роботи програми

```
Введіть пароль: 15
False
```

Завдання №3

3. Напишіть функцію `is_one_away(word1, word2)`, яка приймає як аргументи два слова `word1` і `word2` і повертає значення `True`, якщо слова мають однакову довжину і відрізняються рівно в 1 символі і `False` в іншому випадку.

Лістинг програми

```
def is_one_away(word1, word2):
    a = 0
    for i in range(len(word1)):
        if word1[i] != word2[i]:
            a += 1
    return len(word1) == len(word2) and a == 1
txt1 = input()
txt2 = input()
print(is_one_away(txt1, txt2))
```

Результати роботи програми

cake	wake
dake	neke
True	False

Завдання №4

4. Напишіть функцію `solve(a, b, c)`, яка приймає як аргументи три цілих числа `a`, `b`, `c` – коефіцієнти квадратного рівняння $ax^2+bx+c=0$, і повертає його корені порядку зростання.

Лістинг програми

```
def solve(a, b, c):
    d = (b ** 2) - 4 * a * c
    x1 = ((-1 * b) - d ** 0.5) / (2 * a)
    x2 = ((-1 * b) + d ** 0.5) / (2 * a)
    return min(x1, x2), max(x1, x2)
a, b, c = int(input()), int(input()), int(input())
x1, x2 = solve(a, b, c)
print(x1, x2)
```

Результати роботи програми

1	1	-2
-4	2	7
-5	1	-5
-1.0 5.0	-1.0 -1.0	1.0 2.5

Завдання №5

Опис проекту: програма генерує випадкове число в діапазоні від 1 до 100 і просить користувача вгадати це число. Якщо здогад користувача більше випадкового числа, то програма повинна вивести повідомлення 'Занадто багато, спробуйте ще раз'. Якщо здогад менше випадкового числа, то програма повинна вивести повідомлення 'Занадто мало, спробуйте ще раз'. Якщо користувач вгадає число, програма повинна привітати його та вивести повідомлення 'Ви вгадали, вітаємо!'.

Користувач потенційно може ввести невірні дані, наприклад, не число, або число, що перевищує 100. Важливо передбачити таку можливість, щоб програма продовжувала працювати правильно. Обробка таких ситуацій називається захистом від дурня.

Напишіть функцію `is_valid()`, в яку передається один рядковий аргумент. Функція повертає значення `True` якщо переданий аргумент є цілим числом від 1 до 100 і `False` інакше.

Додайте підрахунок спроб, зроблених користувачем. Коли число відгадано, програма має показати кількість спроб;

Додайте можливість генерації нового числа (повторна гра) після того, як користувач вгадав число;

Додайте можливість вказівки правого кордону для випадкового вибору числа (від 1 до `n`).

Лістинг програми

```
import random
def is_valid(n):
    return True if isinstance(n, (int, float)) or 0 < n < 100 else False

def validate_range(n):
    return n in range(1, 100)

def input_num():
    try:
        n = int(input("Введіть число у проміжку(1,100): "))
    except ValueError as e:
        print(f"Ви ввели неправильний тип даних, ваше число буде рандомно  
вибрано")
        n = random.randint(1, 100)
    while not validate_range(n):
        print("Ви вийшли за рамки проміжку, проміжок від 1 до 100!!!")
        n = int(input("Введіть число у проміжку(1,100): "))
    return n

def play():
    mistake_count = 0
    global answer

    n = input_num()
    guess_number = random.randint(1, n)
    try:
```

```

        answer = int(input("Загадане число: "))
    except not is_valid(answer) or TypeError as e:
        raise e
    while answer != guess_number:
        if answer > guess_number:
            print('Занадто багато, спробуйте ще раз ')
        elif answer < guess_number:
            print('Занадто мало, спробуйте ще раз ')
        mistake_count += 1
        answer = int(input(f"Загадане число: ви зробили {mistake_count} помилку
"))
    print(f'Ви вгадали, вітаємо! число {guess_number}, кількість помилок =
{mistake_count}')
    play_again = input("Зіграємо ще раз? yes/no ")
    return True if play_again == 'yes' else False

def run():
    while True:
        if not play():
            print("Приходьте ще! ")
            break
run()

```

Результати роботи програми

```

Введіть число у проміжку(1,100): 5
Загадане число: 1
Занадто мало, спробуйте ще раз
Загадане число: ви зробили 1 помилку 3
Занадто мало, спробуйте ще раз
Загадане число: ви зробили 2 помилку 2
Занадто мало, спробуйте ще раз
Загадане число: ви зробили 3 помилку 4
Ви вгадали, вітаємо! число 4, кількість помилок = 3
Зіграємо ще раз? yes/no no
Приходьте ще!

```

```

Введіть число у проміжку(1,100): 500
Ви вийшли за рамки проміжку, проміжок від 1 до 100!!
Введіть число у проміжку(1,100): 3
Загадане число: 2
Занадто мало, спробуйте ще раз
Загадане число: ви зробили 1 помилку 1
Занадто мало, спробуйте ще раз
Загадане число: ви зробили 2 помилку 3
Ви вгадали, вітаємо! число 3, кількість помилок = 2
Зіграємо ще раз? yes/no

```

Висновок

Застосовано на практиці використання функцій у мові програмування Python.