Теоретическая часть

Типы ошибок, обнаруженных в задачах

- Логическая ошибка (в диапазоне range при расчете факториала)

- Ошибка выхода за границы массива (в цикле for с диапазоном len + 1)

- Неправильная проверка условий (работа с условными точками останова)

Пояснение к условным точкам останова

Условные точки останова позволяют остановить выполнение программы только при выполнении определенного условия. Это удобно при отладке, когда нужно проанализировать поведение программы при конкретных входных данных, например: password == "qwerty".

Задача 1: Расчет факториала

Код с ошибкой:

def factorial(n):  
 result = 1  
 for i in range(1, n):  
 result \*= i  
 return result  
print(factorial(5)) # 24

Ошибка: range(1, n) не включает n.

Исправленный код:

def factorial(n):  
 result = 1  
 for i in range(1, n + 1):  
 result \*= i  
 return result  
print(factorial(5)) # 120

Задача 2: Проверка пароля

Код:

def check\_password(password):  
 if len(password) < 8:  
 return "Слишком короткий!"  
 elif not any(char.isdigit() for char in password):  
 return "Нет цифр!"  
 else:  
 return "Пароль надёжен!"  
print(check\_password("qwerty"))

Условная точка останова установлена на строке с if при условии password == "qwerty".

Задача 3: Сумма массива

Код с ошибкой:

def calculate\_sum(arr):  
 total = 0  
 for i in range(0, len(arr) + 1):  
 total += arr[i]  
 return total  
numbers = [10, 20, 30]  
print(calculate\_sum(numbers))

Ошибка: выход за пределы массива при доступе по индексу.

Исправленный код:

def calculate\_sum(arr):  
 total = 0  
 for i in range(0, len(arr)):  
 total += arr[i]  
 return total  
numbers = [10, 20, 30]  
print(calculate\_sum(numbers)) # 60

Заключение

В ходе выполнения заданий были исправлены логические ошибки, использованы точки останова и проведён пошаговый анализ выполнения программ. Это позволило детально понять, как работает отладчик и как с его помощью улучшать качество кода.