**Lab Manual for Python Projects**

---

**1. ATM Money Distribution**

**Aim**:
To simulate ATM cash withdrawal in different denominations.

**Algorithm**:

1. Input the withdrawal amount.
2. Calculate the number of ₹2000, ₹500, and ₹100 notes.
3. Output the count for each denomination.

**Code**:

```
amount = int(input("Enter amount to withdraw: "))

notes_2000 = amount // 2000

amount %= 2000

notes_500 = amount // 500

amount %= 500

notes_100 = amount // 100


print(f"2000s: {notes_2000}, 500s: {notes_500}, 100s: {notes_100}")
```

**Output**:

```
Enter amount to withdraw: 3700

2000s: 1, 500s: 3, 100s: 2
```

**Result**:
The program successfully simulates ATM money distribution.

---

**2. Sequential Removal**

**Aim**:
To remove elements from a sequence based on specific rules.

**Algorithm**:

1. Input a sequence and the removal criteria.
2. Remove elements sequentially based on the rules.

3. Output the modified sequence.

**Code**:

```
sequence = input("Enter a sequence: ")

remove = input("Enter characters to remove: ")


for char in remove:

   sequence = sequence.replace(char, "")


print("Modified sequence:", sequence)
```

**Output**:

Enter a sequence: hello

Enter characters to remove: eo

Modified sequence: hll

**Result**:
The program successfully removes elements from the sequence based on rules.

---

**3. String Validation II**

**Aim**:
To validate a string for uppercase, digits, and special characters.

**Algorithm**:

1. Input a string.
2. Check if the string contains uppercase letters, digits, and special characters.
3. Output validation results.

**Code**:

```
import re


def validate_string(s):

   if re.search("[A-Z]", s) and re.search("[0-9]", s) and re.search("[!@#$%&*]", s):

      print("String is valid")
```

```
    else:
        print("String is invalid")


validate_string(input("Enter a string: "))
```

**Output**:

```
Enter a string: Hello123!
String is valid
```

**Result**:
The string validation is successfully performed.

---

**4. Number Operations**

**Aim**:
To perform basic arithmetic operations on user inputs.

**Algorithm**:

1. Input two numbers.
2. Perform addition, subtraction, multiplication, and division.
3. Output the results.

**Code**:

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))


print(f"Addition: {a + b}")
print(f"Subtraction: {a - b}")
print(f"Multiplication: {a * b}")
print(f"Division: {a / b}")
```

**Output**:

```
Enter first number: 10
Enter second number: 5
Addition: 15
```

Subtraction: 5

Multiplication: 50

Division: 2.0

**Result**:
The program performs the basic arithmetic operations correctly.

---

**5. Duplicate Number**

**Aim**:
To find and handle duplicate numbers in a list.

**Algorithm**:

1. Input a list of numbers.
2. Identify and count duplicates.
3. Output the result.

**Code**:

```
nums = list(map(int, input("Enter numbers: ").split()))

duplicates = set([x for x in nums if nums.count(x) > 1])


print("Duplicates:", duplicates)
```

**Output**:

Enter numbers: 1 2 3 4 5 3 2

Duplicates: {2, 3}

**Result**:
The program identifies duplicate numbers successfully.

---

**6. Intersection of Two Arrays**

**Aim**:
To find common elements between two arrays.

**Algorithm**:

1. Input two arrays.
2. Find the intersection of both arrays.

3. Output the common elements.

**Code**:

```
arr1 = list(map(int, input("Enter first array: ").split()))

arr2 = list(map(int, input("Enter second array: ").split()))


intersection = set(arr1) & set(arr2)


print("Intersection:", intersection)
```

**Output**:

Enter first array: 1 2 3 4 5

Enter second array: 3 4 5 6 7

Intersection: {3, 4, 5}

**Result**:
The program correctly finds the intersection between two arrays.

---

**7. First Unique Character**

**Aim**:
To find the first non-repeating character in a string.

**Algorithm**:

1. Input a string.
2. Check each character for uniqueness.
3. Output the first unique character.

**Code**:

```
string = input("Enter a string: ")


for char in string:
   if string.count(char) == 1:
      print("First unique character:", char)
      break
```

**Output**:

Enter a string: swiss

First unique character: w

**Result**:
The program finds the first unique character successfully.

---

**8. Reverse Vowels**

**Aim**:
To reverse the vowels in a string.

**Algorithm**:

1. Input a string.
2. Identify and reverse the vowels.
3. Output the modified string.

**Code**:

```
def reverse_vowels(s):
    vowels = [c for c in s if c in "aeiouAEIOU"]
    result = []
    for c in s:
        if c in "aeiouAEIOU":
            result.append(vowels.pop())
        else:
            result.append(c)
    return ''.join(result)


string = input("Enter a string: ")
print("Modified string:", reverse_vowels(string))
```

**Output**:

Enter a string: hello

Modified string: holle

**Result**:
The vowels are successfully reversed.

---

**9. Twin Prime**

**Aim**:
To find twin prime numbers (pairs of primes that differ by 2).

**Algorithm**:

1. Input a range.
2. Identify twin primes within the range.
3. Output the twin primes.

**Code**:

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True


def find_twin_primes(start, end):
    for num in range(start, end):
        if is_prime(num) and is_prime(num + 2):
            print(num, num + 2)


start = int(input("Enter start: "))
end = int(input("Enter end: "))
find_twin_primes(start, end)
```

**Output**:

```
Enter start: 1
```

Enter end: 20

3 5

5 7

11 13

17 19

**Result**:
The program successfully identifies twin prime numbers.

---

**10. Electricity Bill**

**Aim**:
To calculate electricity bills based on usage.

**Algorithm**:

1. Input the number of units consumed.
2. Calculate the total bill based on tariff.
3. Output the bill amount.

**Code**:

```
units = int(input("Enter units consumed: "))


if units <= 100:
   bill = units * 1.5
elif units <= 200:
   bill = 100 * 1.5 + (units - 100) * 2
else:
   bill = 100 * 1.5 + 100 * 2 + (units - 200) * 3


print(f"Total bill: {bill}")
```

**Output**:

Enter units consumed: 250

Total bill: 575.0

**Result**:
The program calculates the electricity bill correctly.

---

**11. Discounted Price**

**Aim**:
To compute the price after applying a discount.

**Algorithm**:

1. Input the original price and discount percentage.
2. Calculate the discounted price.
3. Output the final price.

**Code**:

```
price = float(input("Enter original price: "))

discount = float(input("Enter discount percentage: "))


final_price = price - (price * (discount / 100))


print(f"Final price after discount: {final_price}")
```

**Output**:

yaml

Copy code

Enter original price: 1000

Enter discount percentage: 10

Final price after discount: 900.0

**Result**:
The program correctly computes the discounted price.

---

**12. Name Password Generation**

**Aim**:
To create a secure password using a person's name.

**Algorithm**:

1. Input a name.
2. Generate a password by modifying characters (e.g., replace vowels with numbers).
3. Output the generated password.

**Code**:

python

Copy code

```
name = input("Enter your name: ")

password = name.replace('a', '@').replace('e', '3').replace('i', '1').replace('o', '0').replace('u', '7')

print("Generated password:", password)
```

**Output**:

yaml

Copy code

```
Enter your name: JohnDoe

Generated password: J0hnD03
```

**Result**:
The password is generated successfully based on the name.

---

**13. Student Attendance Record**

**Aim**:
To track and analyze student attendance.

**Algorithm**:

1. Input student attendance records.
2. Analyze the records to determine presence or absence.
3. Output the attendance summary.

**Code**:

```
students = int(input("Enter number of students: "))

attendance = []
```

```
for i in range(students):
    name = input(f"Enter name of student {i+1}: ")
    present = input(f"Is {name} present (Y/N)? ")
    attendance.append((name, present == 'Y'))

print("\nAttendance Summary:")
for name, present in attendance:
    status = "Present" if present else "Absent"
    print(f"{name}: {status}")
```

**Output**:

```
Enter number of students: 2
Enter name of student 1: John
Is John present (Y/N)? Y
Enter name of student 2: Jane
Is Jane present (Y/N)? N

John: Present
Jane: Absent
```

**Result**:
The program successfully tracks and outputs student attendance.