```
#Question 1
def word_frequency(input_string):
    word_dict = {}
    words = input_string.lower().split()

    for word in words:
        word = word.strip(".,?!:;\"'")
        word_dict[word] = word_dict.get(word, 0) + 1

    return word_dict


# Test the function
input_string = "This is a test string. Test is a word in this string."
print(word_frequency(input_string))

#Question 2
def is_palindrome(input_string):
    return input_string.lower() == input_string.lower()[::-1]

def count_vowels(input_string):
    vowels = "aeiou"
    return sum(1 for char in input_string.lower() if char in vowels)

# Test the program
input_string = "Madam"
if is_palindrome(input_string):
    print(f"The input string '{input_string}' is a palindrome.")
    print(f"The number of vowels in the string is: {count_vowels(input_string)}")
else:
    print("The input string is not a palindrome.")

#Question 3
def calculate_denominations(amount):
    denominations = [2000, 500, 200, 100, 50, 20, 10, 5, 2, 1]
```

```
    notes_count = {}
    for denomination in denominations:
        count = amount // denomination
        if count > 0:
            notes_count[denomination] = count
            amount %= denomination
    return notes_count

# Test the program
amount = int(input("Enter the amount: "))
denominations_count = calculate_denominations(amount)
print("Count of currency:")
for denomination, count in denominations_count.items
():
    print(f"{denomination} : {count}")

Question 4
def collatz_sequence(n):
    sequence = [n]
    while n != 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
        sequence.append(n)
    return sequence

def find_longest_chain(limit):
    max_length = 0
    start_number = 0
    for i in range(1, limit):
        sequence_length = len(collatz_sequence(i))
        if sequence_length > max_length:
            max_length = sequence_length
            start_number = i
    return start_number

# Test the function
limit = 1000000
starting_number = find_longest_chain(limit)
print(f"The starting number with the longest chain
```

```
less than {limit} is: {starting_number}")

Question 5
class ATM:
    def __init__(self):
        self.users = {}
        self.account_counter = 1000  # Starting account number

    def create_user_account(self, name, initial_balance):
        account_number = self.account_counter
        self.account_counter += 1
        self.users[account_number] = {'name': name, 'balance': initial_balance}
        return account_number

    def check_balance(self, account_number):
        if account_number in self.users:
            user = self.users[account_number]
            print(f"Account Holder Name: {user['name']}")
            print(f"Account Balance: {user['balance']}")
        else:
            print("Account not found.")

    def withdraw_money(self, account_number, amount):
        if account_number in self.users:
            user = self.users[account_number]
            if amount <= user['balance']:
                user['balance'] -= amount
                print("Withdrawal successful.")
            else:
                print("Insufficient balance.")
        else:
            print("Account not found.")

    def deposit_money(self, account_number, amount):
        if account_number in self.users:
```

```
110             user = self.users[account_number]
111             user['balance'] += amount
112             print("Deposit successful.")
113         else:
114             print("Account not found.")
115 
116 
117 def main():
118     atm = ATM()
119 
120     while True:
121         print("\nATM Application")
122         print("1. Create User Account")
123         print("2. Check Balance")
124         print("3. Withdraw Money")
125         print("4. Deposit Money")
126         print("5. Exit")
127 
128         choice = int(input("Enter your choice (1-5
    ): "))
129 
130         if choice == 1:
131             name = input("Enter your name: ")
132             initial_balance = float(input("Enter
    initial balance: "))
133             account_number = atm.create_user_account
    (name, initial_balance)
134             print(f"Account created successfully.
    Your account number is: {account_number}")
135 
136         elif choice == 2:
137             account_number = int(input("Enter your
    account number: "))
138             atm.check_balance(account_number)
139 
140         elif choice == 3:
141             account_number = int(input("Enter your
    account number: "))
142             amount = float(input("Enter the amount
    to withdraw: "))
143             atm.withdraw_money(account_number,
```

```
amount)

        elif choice == 4:
            account_number = int(input("Enter your account number: "))
            amount = float(input("Enter the amount to deposit: "))
            atm.deposit_money(account_number, amount)

        elif choice == 5:
            print("Exiting the ATM Application. Have a nice day!")
            break

        else:
            print("Invalid choice. Please try again.")


if __name__ == "__main__":
    main()

```