1. Opisz MT, która dodaje 1 do liczby binarnej zapisanej na taśmie. Napisz w dowolnym języku program, który realizuje ten algorytm.

   Opis:

   - Rozpocznij na ostatnim bicie liczby binarnej na taśmie.
   - Jeśli bit jest równy 0, zapisz 1 na taśmie w miejscu tego bitu, zatrzymaj się i zakończ działanie maszyny.
   - Jeśli bit jest równy 1, zapisz 0 na taśmie w miejscu tego bitu i przesuń się o jedno pole w lewo.
   - Powtarzaj kroki 2-3, dopóki nie napotkasz bitu o wartości 0, co oznacza, że dodałeś 1 do liczby binarnej.

```python
# add_one_to_binary_tape.py
def add_one_to_binary_tape(binary_tape):
    tape = list(binary_tape)
    carry = 1
    index = len(tape) - 1

    while index >= 0:
        if tape[index] == '0':
            tape[index] = '1'
            carry = 0
            break
        else:
            tape[index] = '0'
            index -= 1

    if carry == 1:
        tape.insert(0, '1')

    return ''.join(tape)

# Przykład użycia:
binary_number = "1101"
result = add_one_to_binary_tape(binary_number)
print(result)  # Wynik to "1110"
```

2. Opisz MT, która oblicza ilość 1 w swoim wejściowym słowie.
    - Rozpocznij na pierwszym bicie wejściowego słowa na taśmie.
    - Przejdź przez każdy bit w słowie, sprawdzając, czy jest to "1".
    - Jeśli bit jest "1", zwiększ licznik o 1.
    - Przesuń się o jedno pole w prawo.
    - Powtarzaj kroki 2-4, dopóki nim przejdziesz przez całe słowo.
    - Po przejściu przez całe słowo, zatrzymaj się i zwróć wartość licznika jako wynik.

    "1101011" -> 5

3. Opisz maszyne turinga, który zastępuje w dowolnym tekście zbudowanym z liter A, B, C i D wszystkie literki A literką C. Na przykład z tekstu BACA ma powstać BCCC.
    - Rozpocznij na pierwszym bicie wejściowego słowa na taśmie.
    - Przejdź przez każdy bit w słowie, sprawdzając, czy jest to "A"
    - Jeżeli jest zastąp literą „C”.
    - Przesuń się o jedno pole w prawo.
    - Powtarzaj kroki 2-4, dopóki nim przejdziesz przez całe słowo.
    - Po przejściu przez całe słowo, zatrzymaj się i zwróć wartość taśme jako słowo.

```python
def replace_a_with_c(input_string):
    tape = list(input_string)
    index = 0

    while index < len(tape):
        if tape[index] == 'A':
            tape[index] = 'C'
        index += 1

    return ''.join(tape)

# Przykład użycia:
input_text = "BACA"
result = replace_a_with_c(input_text)
print(result)  # Wynik to "BCCC"
```

4. Opisz maszyne turinga, który przenosi pierwszą literkę wyrazu zbudowanego z liter A, B, C i D z początku na koniec. Na przykład z tekstu DAB ma powstać ABD.

    - Rozpocznij na pierwszej literze (najbardziej na lewo) w tekście na taśmie.
    - Odczytaj tę literę.
    - Przesuń się o jedno pole w lewo i zapisz odczytaną literę na tym nowym miejscu.
    - Przesuń się na koniec tekstu.
    - Zapisz wcześniej odczytaną literę na nowym miejscu na końcu tekstu.
    - Zatrzymaj się i zakończ działanie maszyny.

```python
def move_first_letter_to_end(input_text):
    tape = list(input_text)
    if len(tape) > 1:
        first_letter = tape[0]
        tape[0] = tape[1]

        for i in range(1, len(tape) - 1):
            tape[i] = tape[i + 1]

        tape[-1] = first_letter

    return ''.join(tape)


# Przykład użycia:
input_text = "ABCDEF"
result = move_first_letter_to_end(input_text)
print(result)  # Wynik to "BCDEFA"
```

5. Napisz maszyne turinga który zamienia miejscami początkowy i końcowy bit liczby binarnej. Na przykład liczba 11010 powinna przejść w 01011.
    - Rozpocznij na pierwszym bicie (najbardziej znaczącym) liczby binarnej na taśmie.
    - Odczytaj wartość tego bitu.
    - Przesuń się na koniec liczby binarnej (najmniej znaczący bit).
    - Odczytaj wartość tego bitu.
    - Zamień odczytane bity miejscami.
    - Zapisz zmodyfikowaną liczbę na taśmie.
    - Zatrzymaj się i zakończ działanie maszyny.

```python
def swap_start_and_end_bits(binary_number):
    tape = list(binary_number)
    if len(tape) >= 2:
        start_bit = tape[0]
        end_bit = tape[-1]

        tape[0] = end_bit
        tape[-1] = start_bit

    return ''.join(tape)

# Przykład użycia:
binary_number = "11010"
result = swap_start_and_end_bits(binary_number)
print(result)  # Wynik to "01011"
```