1. Nasabah Bank

Di sebuah bank, setiap nasabah yang datang mengambil nomor antrian dan menunggu giliran untuk dilayani sesuai urutan kedatangan. Buatlah simulasi antrian di bank menggunakan konsep queue.

Setiap nasabah memiliki beberapa pilihan transaksi yang dapat dilakukan:

1. **Tarik Tunai** - mengurangi saldo nasabah jika saldo mencukupi.
2. **Cek Saldo** - menampilkan saldo saat ini.
3. **Setor Tunai** - menambah saldo nasabah.

Program ini akan meminta pengguna untuk:

1. Memasukkan jumlah nasabah.
2. Memasukkan nama masing-masing nasabah.
3. Memilih transaksi untuk setiap nasabah dalam antrian hingga antrian selesai.

Input:

```
Masukkan jumlah nasabah: 2
Masukkan nama nasabah 1: Andi
Masukkan nama nasabah 2: Budi
```

Output:

```
Nasabah Andi sedang dilayani.

Pilih transaksi:
1. Tarik Tunai
2. Cek Saldo
3. Setor Tunai
4. Selesai
Masukkan nomor transaksi: 2
Saldo Andi: Rp1000000

Pilih transaksi:
1. Tarik Tunai
2. Cek Saldo
3. Setor Tunai
4. Selesai
Masukkan nomor transaksi: 1
Masukkan jumlah yang ingin ditarik: 200000
Andi berhasil menarik tunai Rp200000. Saldo sekarang: Rp800000
```

```
Pilih transaksi:
1. Tarik Tunai
2. Cek Saldo
3. Setor Tunai
4. Selesai
Masukkan nomor transaksi: 4
Andi telah selesai melakukan transaksi.

Nasabah Budi sedang dilayani.
Pilih transaksi:
1. Tarik Tunai
2. Cek Saldo
3. Setor Tunai
4. Selesai
Masukkan nomor transaksi: 4
Budi telah selesai melakukan transaksi.

Antrian Kali ini Telah Selesai
```

2. Cek Palindrom dengan Queue dan Stack (Fitur Pengolah Kata)

Deskripsi:

Kamu diminta untuk membuat fitur dalam pengolah kata (word processor) yang dapat memeriksa apakah sebuah kata merupakan palindrome atau tidak. Untuk melakukannya, gunakan **Queue** dan **Stack**.

Sebuah kata dikatakan **palindrome** jika kata tersebut dibaca sama baik dari depan maupun dari belakang, contohnya: "radar", "madam", atau "level".

Implementasi:

1. Gunakan **Queue** untuk menyimpan setiap karakter kata dalam urutan aslinya (dari depan ke belakang).
2. Gunakan **Stack** untuk menyimpan karakter yang sama, tetapi dalam urutan terbalik (dari belakang ke depan).

3. Periksa apakah karakter yang keluar dari **Queue** sama dengan karakter yang keluar dari **Stack**. Jika semua karakter cocok, maka kata tersebut adalah palindrome.

Spesifikasi:

- Hanya gunakan fungsi-fungsi Queue dan Stack untuk implementasi solusi ini.

Contoh Input dan Output:

Contoh Input 1:

```
word = "radar"
```

Contoh Output 1:

```
True
```

Contoh Input 2:

```
word = "hello"
```

Contoh Output 2:

```
False
```