

Queue

Farah Zakiyah Rahmanti

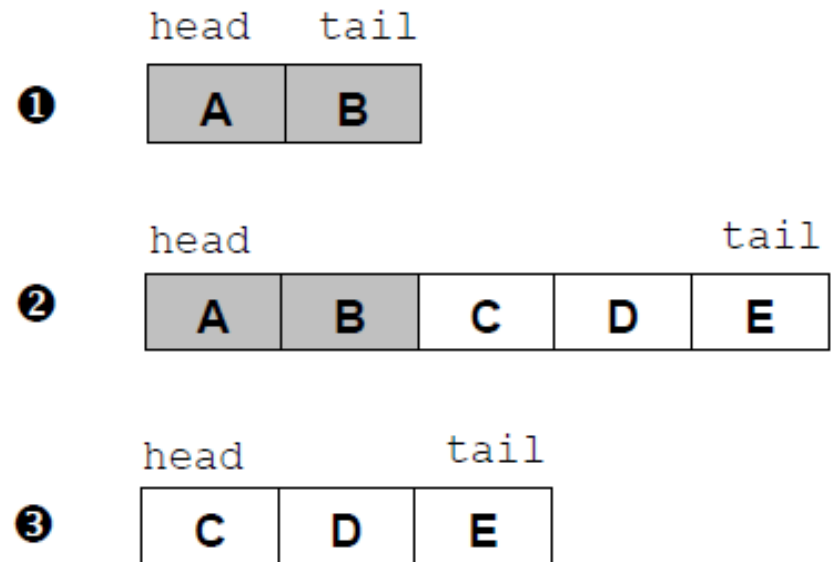
Description

- FIFO (First In First Out)
- How to save and take data with FIFO
- Queue is “antrian” in bahasa indonesia
- Let’s imagine about queue at bank teller



Enqueue - Dequeue

- Queue with 2 elements
- After enqueue C,D,E
- After dequeue A and B



```
#include <stdio.h>
#include <stdlib.h>
#define MAXQUEUE 100;
typedef int ItemType;
typedef struct{
    int    Count;
    int    Front;
    int    Rear;
    ItemType  Item[MAXQUEUE];
}Queue;
void InitializeQueue (Queue *Q)
{
    Q->Count = 0;
    Q->Front = 0;
    Q->Rear = 0;
```

```
int Empty(Queue *Q)
{
    return(Q->Count == 0);
}

int Full(Queue *Q)
{
    return(Q->Count == MAXQUEUE);
}
```

```
void Insert(ItemType ins, Queue *Q)
{
    if (Q->Count == MAXQUEUE)
        printf("Tidak dapat memasukkan data! Queue Penuh!");
    else {
        Q->Item[Q->Rear] = ins;
        Q->Rear = (Q->Rear + 1) % MAXQUEUE;
        ++(Q->Count);
    }
}
```

```
void Remove(Queue *Q, ItemType *rm)
{
    if (Q->Count == 0)
        printf("Tidak dapat mengambil data! Queue Kosong!");
    else {
        *rm = Q->Item[Q->Front];
        Q->Front = (Q->Front + 1) % MAXQUEUE;
        --(Q->Count);
    }
}
```

```
Front = (Front + 1) % N;
```

```
Rear = (Rear + 1) % N;
```

- modular arithmetic process that is applied to the queue

- Try to implement Queue !