

# Array, Stack & Queue

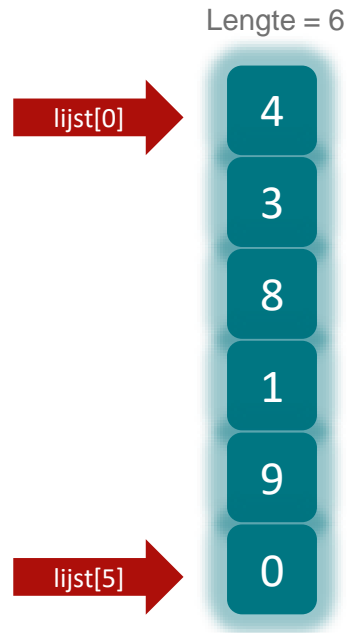
Bachelor IT

Sven Mariën

(sven.marien01@ap.be)

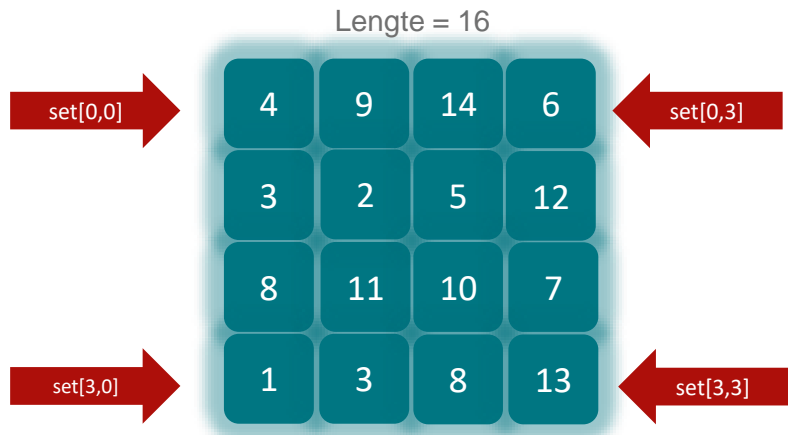
# Arrays

- Ik heb een lijst met getallen, woorden...
- Snel een element opzoeken adhv. zijn 'index'
- 1-dimensie
- Volgorde van de elementen blijft behouden
- Syntax:
  - `int[ ] lijst = new int[6];`
  - `int[ ] lijst = new int[ ] {4,3,8,1,9,0};`



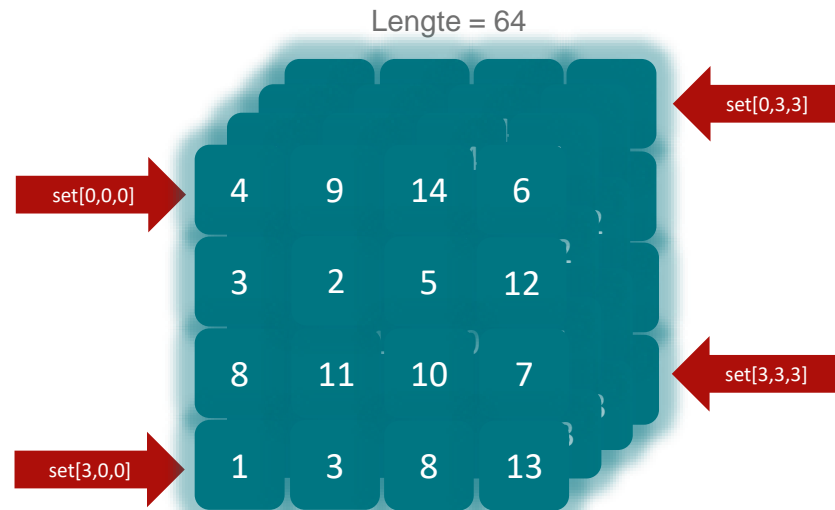
# Arrays

- Ik heb een set van getallen, woorden,...
- Bestaande uit rijen en kolommen
- 2-dimensies
- Syntax:
  - `int[,] set = new int[4,4];`



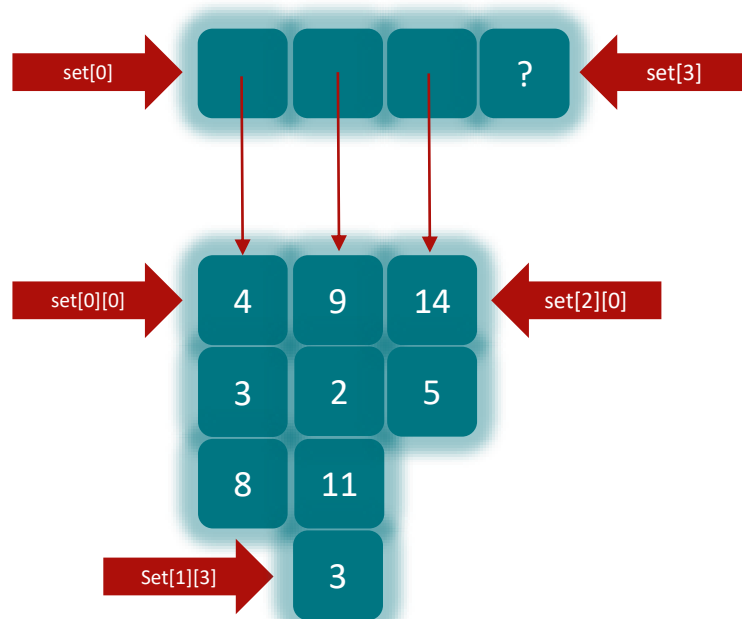
# Arrays

- Mogelijk met 3, 4,... n-dimensies...
- Syntax :
  - `int[,,] set = new int[4,4,4];`



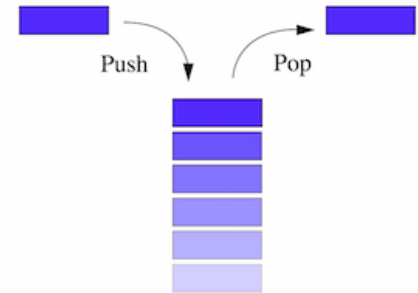
# Array van Arrays

- Ook wel “jagged-array” genaamd.
- Handig indien de 2<sup>e</sup> dimensie geen vaste aantallen heeft
- Syntax:
  - `int[ ][ ] set = new int [4][ ];`
  - `set[0] = new int[3];`
  - `set[1] = new int[4];`
  - `set[2] = new int[2];`
  - `set[3] = ...`



# Stack

- Een “**stack**” (stapel) is een lineaire datastructuur
- Heeft 1 dimensie
- Wordt ook wel een **LIFO** lijst genoemd
  - Last-In, First-out
- Een element toevoegen: **Push**
- Een element afnemen: **Pop**
- Het bovenste element bekijken: **Peek**
- We kunnen een stack implementeren adhv. een **1D-array**

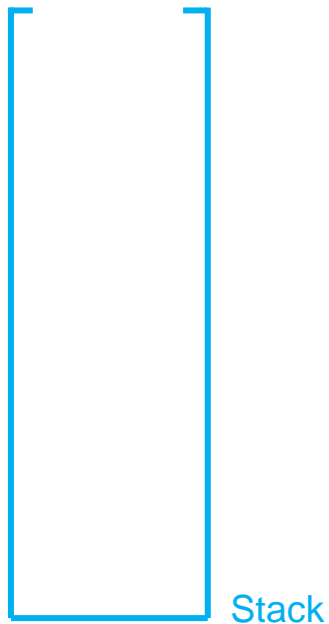


# Werking van de Stack

13

7

4

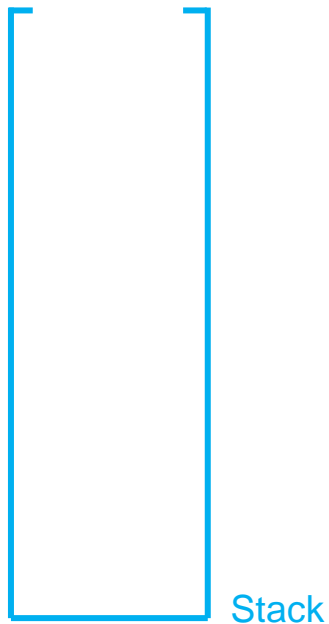


# Werking van de Stack

13

7

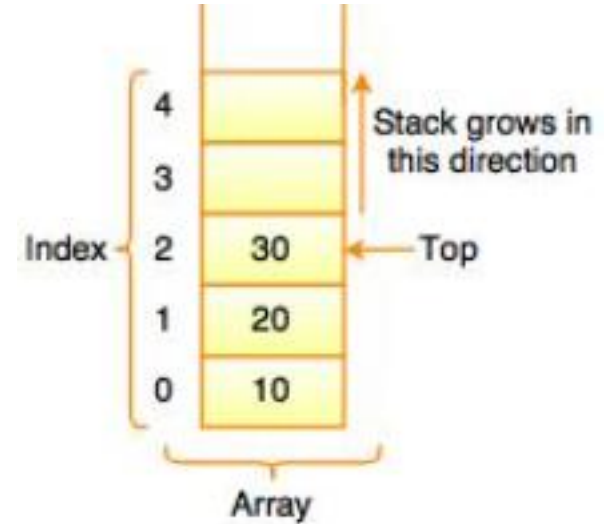
4





# Hoe stack implementeren adhv. array ?

- Array + **“Top” variabele** die de index aangeeft van het bovenste element.
- Deze variabele zal mee verhogen / verlagen bij een push / pop
- Check : Array Leeg / Volzet !

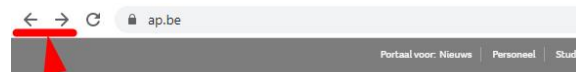


# Toepassingen van een stack

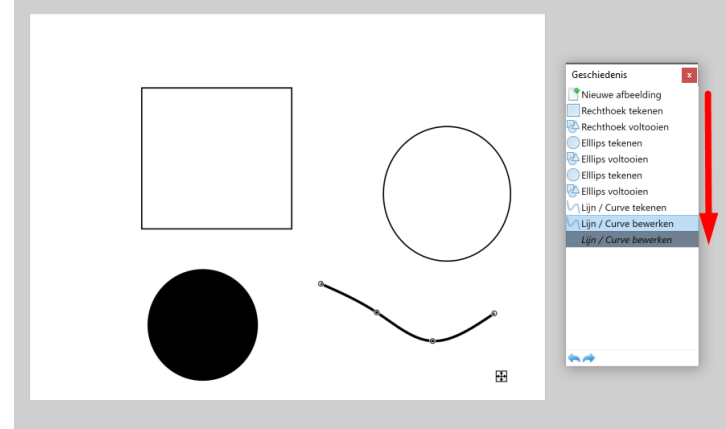
- Undo/Redo
  - Office, Paint,...



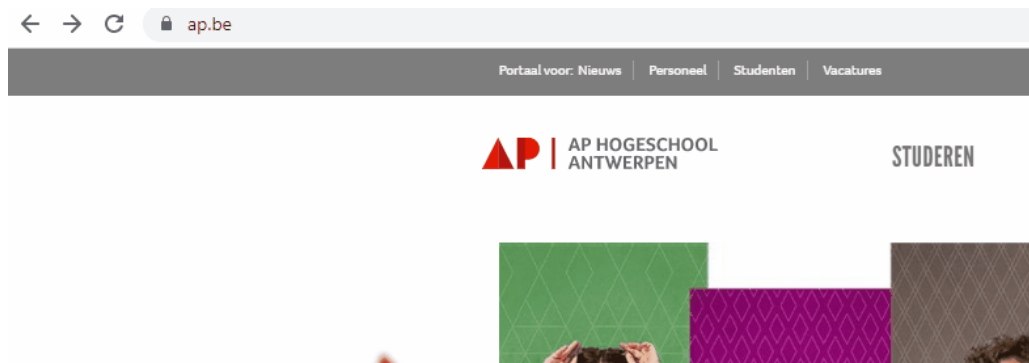
- Back/Forward in browser



AP | AP HOGESCHOOL  
ANTWERPEN



# “Back”-stack in de browser



Huidige url: **ap.be**

Url Stack



# Call-Stack bij het uitvoeren van een c# applicatie

```
0 references
public MainWindow()
{
    InitializeComponent();

    string zin = VerwijderSpaties("Dit Is EEn ZIN MET GROTE en kleine letters en spaties");
}

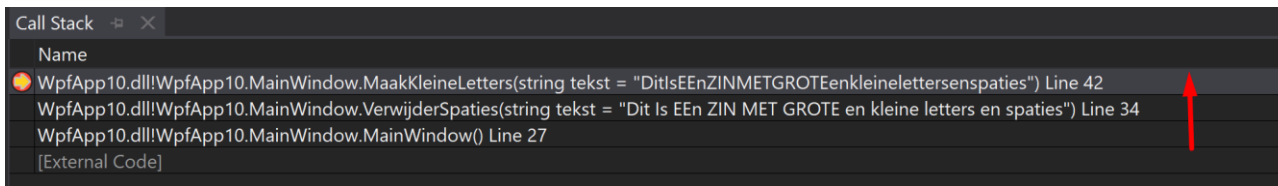
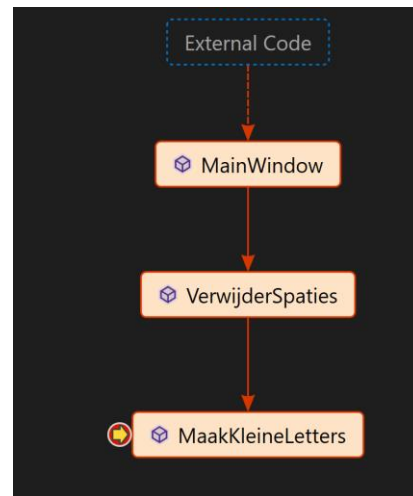
1 reference
string VerwijderSpaties(string tekst)
{
    string TekstZonderSpaties = tekst.Replace(" ", "");

    string TekstZonderSpatiesEnKleineLetters = MaakKleineLetters(TekstZonderSpaties);

    return TekstZonderSpatiesEnKleineLetters;
}

1 reference
string MaakKleineLetters (string tekst)
{
    string KleineLetters = tekst.ToLower();

    return KleineLetters;
}
```



# Stack: Controleren / Berekenen van expressies

- bv:  $\{a^2 - [(b + c)^2 - (d + e)^2] * [\sin(x - y)]\} - \cos(x + y)$
- De haakjes moeten steeds in balans zijn !

{[(())][()]}()

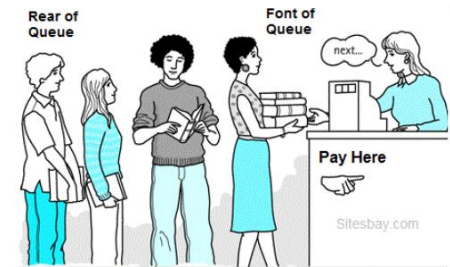
- Kan gecontroleerd worden mbv. een stack
  - Haakje open : push op stack
  - Haakje sluit : pop van stack en vergelijk het type
    - { [ (

## Beperking van een **stack** op basis van **Array** ?

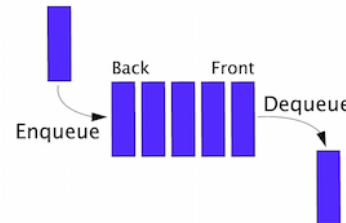
- **Grootte van de array** moet bij de start worden gekozen
  - Array **te groot** in verhouding tot het gebruik => verspilling van geheugen
  - Array **te klein** => stackoverflow
- **Oplossing ?**
- Stack volzet ? -> Nieuwe array aanmaken en alle data over kopiëren
  - => oefeningen

# Queue

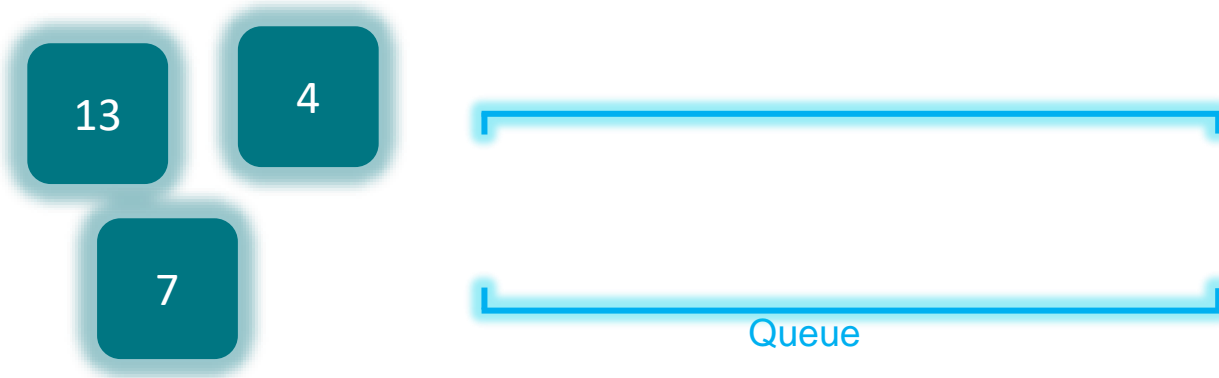
- Een “**queue**” (wachtrij / buffer) is een lijst datastructuur
- Heeft 1 dimensie
- Wordt ook wel een **FIFO** lijst genoemd
  - First-In, First-out
- Toevoegen van een element : **Enqueue**
- Verwijderen van een element : **Dequeue**
- We kunnen een queue ook bouwen adhv. een **1D-array**



Real Life Example of Queue : Library Counter

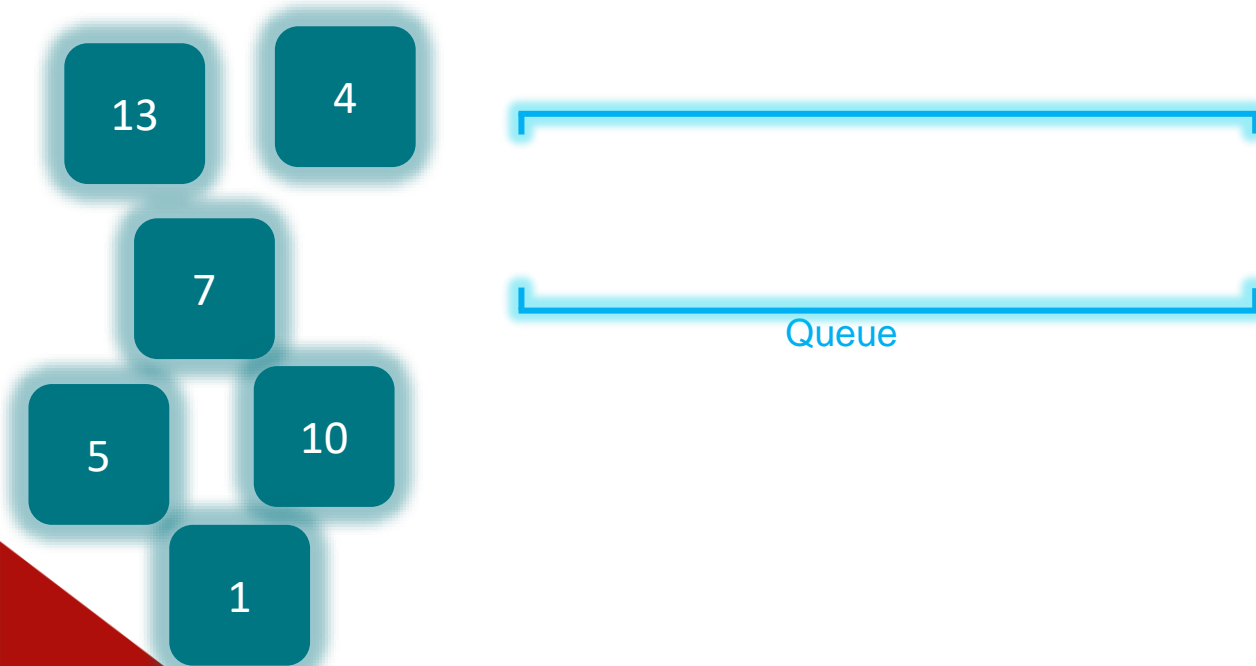


# Werking van de Queue



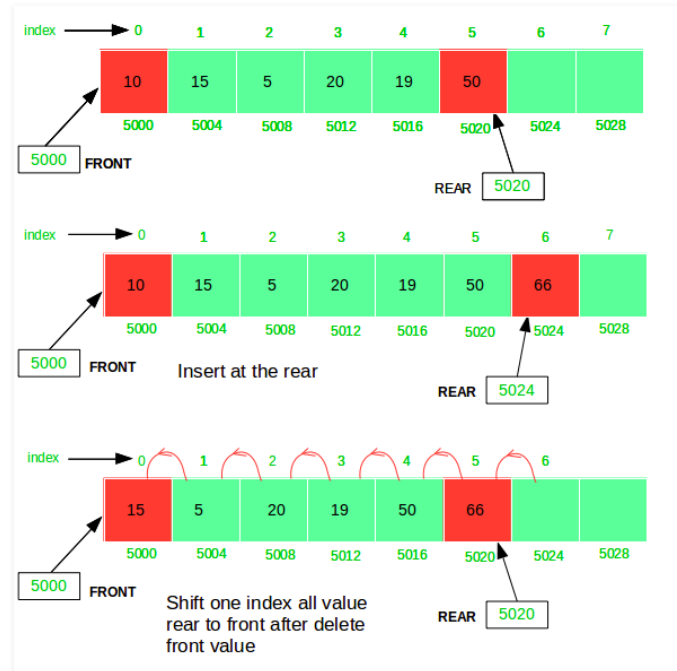


# Werking van de Queue



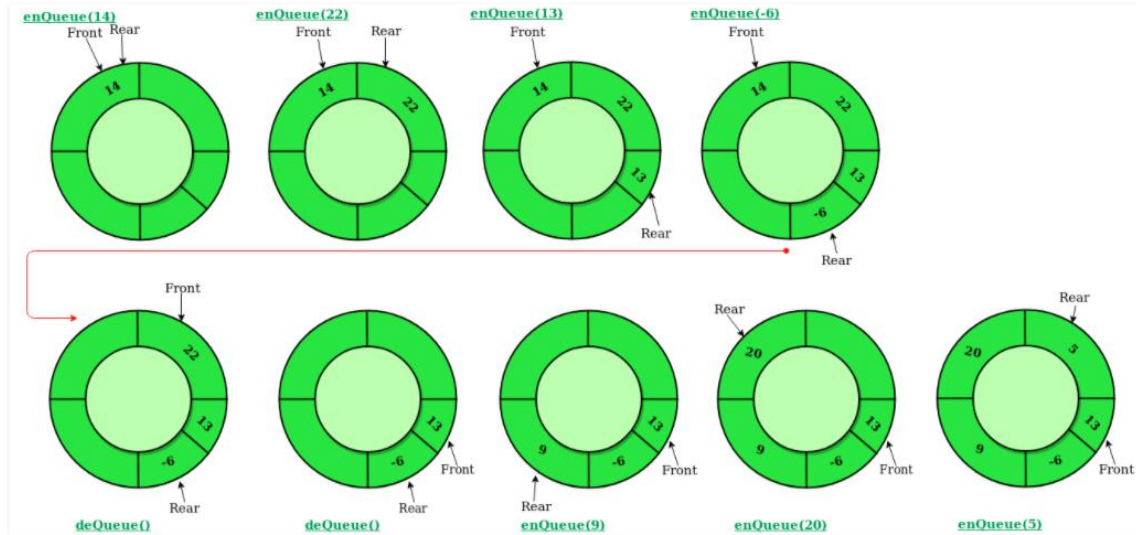
# Hoe queue implementeren mbv. Array ?

- Lineair queue

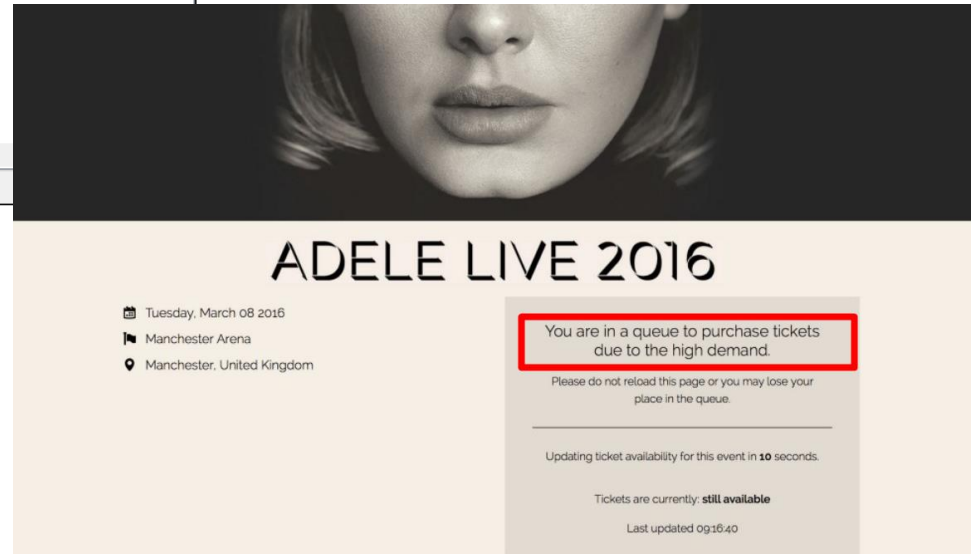
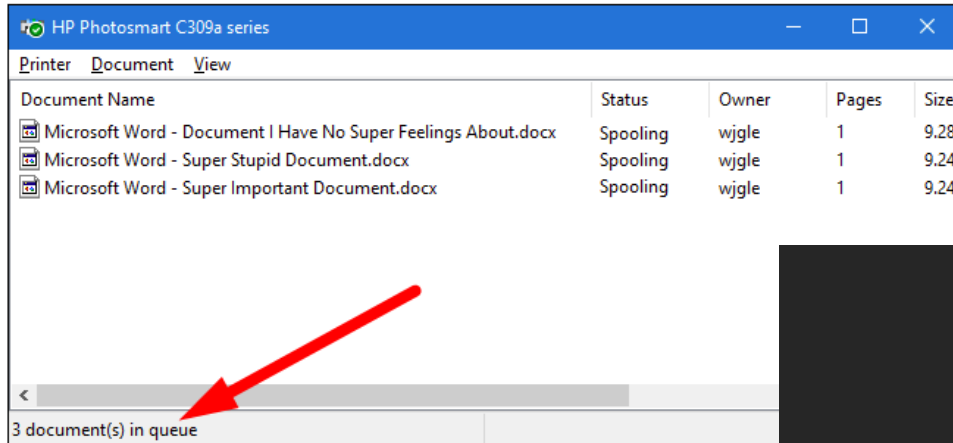


# Hoe queue implementeren mbv. Array ?

- Circular Queue

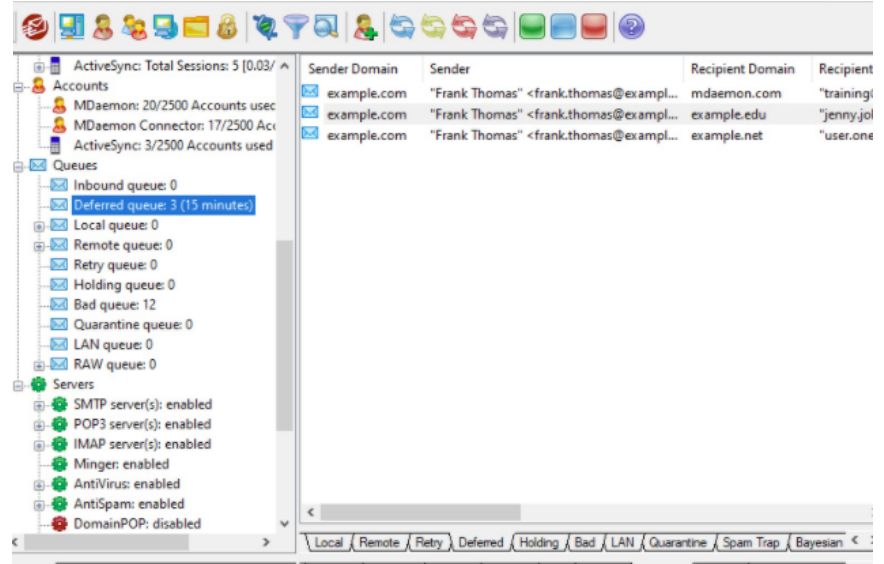
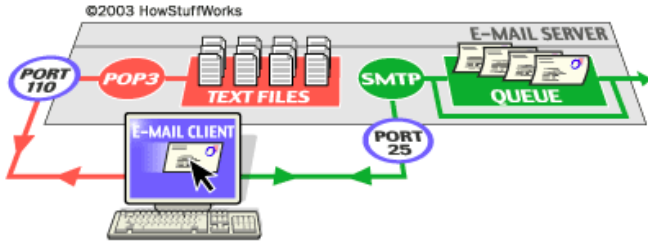


# Toepassing van een queue



# Toepassing van queue

- E-mail versturen, SMS versturen,...



# Beperking van een **queue** op basis van een **array** ?

- **Grootte van de array** moet bij de start worden gekozen
  - Array **te groot** in verhouding tot het gebruik => verspilling van geheugen
  - Array **te klein** => queue full
- **Oplossing ?**
- Queue volgt ?
  - => Nieuwe array aanmaken en alle data over kopiëren
  - Lineaire ⇔ Circulaire queue

