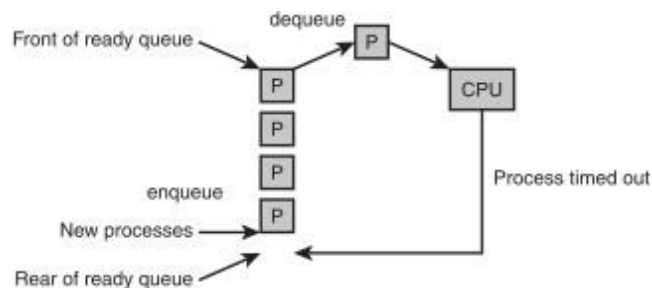


## Modul 2 – *Queue & Deque*

---

1. Buatlah suatu modul dengan nama *queueProgram*, dengan fungsi-fungsi utama yang terdapat dalam modul adalah:
  - `createQueue ( )` : inisialisasi stack kosong
  - `enqueue (data),` : penambahan data baru pada queue
  - `dequeue ( ),` : penghapusan data
  - `isEmpty(),` : pengecekan apakah queue dalam keadaan kosong
  - `size ( ),` : informasi jumlah data yang terdapat pada queue
2. Terdapat proses penjadwalan CPU. Pada penjadwalan CPU terdapat beberapa proses sebagai berikut :
  - a. Semua *task* masuk kedalam antrian, setiap *task* ini terdapat atribut waktu yang dibutuhkan masing-masing *task* untuk menggunakan CPU. Misalkan *task* 'A' membutuhkan waktu untuk diproses di CPU selama 7 detik.
  - b. CPU memiliki atribut waktu, yaitu semua *task* hanya boleh memakai CPU selama waktu tertentu. Misalkan waktu CPU adalah 3 detik, maka semua *task* hanya boleh memakai waktu CPU 3 detik saja.
  - c. *Task* yang berada di posisi paling depan dari antrian, mendapat kesempatan pertama untuk diproses ke dalam CPU.
  - d. Jika waktu proses pada suatu *task*, melebihi waktu proses CPU, maka *task* tersebut dimasukkan kembali ke antrian (posisi *rear*)
  - e. Jika waktu proses pada suatu *task* tidak melebihi waktu proses CPU, maka *task* tersebut dikeluarkan dari antrian, yang berarti *task* tersebut telah selesai diproses oleh CPU.

Ilustrasi penjadwalan di CPU ini, dapat dilihat pada Gambar berikut :



Buatlah ilustrasi penjadwalan CPU tersebut dengan menggunakan modul *queueProgram* yang telah dibuat sebelumnya, dengan ketentuan sebagai berikut :

1. Input berupa jumlah *task* atau proses yang akan diproses pada CPU (petunjuk : gunakan list 2D)
2. Nama proses, beserta atribut waktu yang dibutuhkan masing-masing proses untuk menggunakan CPU
3. Waktu proses CPU

Berikut contoh ilustrasi penjadwalan CPU :

### **Contoh Proses Penjadwalan CPU untuk 10 Proses :**

#### **Input Data Proses dan Waktu Proses :**

Jumlah Proses yang akan dijadwal di CPU = 10

Nama Proses ke-0 : A

Waktu proses : 16

Nama Proses ke-1 : B

Waktu proses : 5

Nama Proses ke-2 : C

Waktu proses : 8

Nama Proses ke-3 : D

Waktu proses : 2

Nama Proses ke-4 : E

Waktu proses : 1

Nama Proses ke-5 : F

Waktu proses : 9

Nama Proses ke-6 : G

Waktu proses : 3

Nama Proses ke-7 : H

Waktu proses : 9

Nama Proses ke-8 : I

Waktu proses : 4

Nama Proses ke-9 : J

Waktu proses :

#### **Print data Proses :**

Antrian Proses :

[[ 'J', 11], [ 'I', 4], [ 'H', 9], [ 'G', 3], [ 'F', 9], [ 'E', 1], [ 'D', 2], [ 'C', 8], [ 'B', 5], [ 'A', 16]]

#### **Input waktu proses CPU :**

waktu proses CPU =

## Proses Penjadwalan CPU untuk 10 Proses seperti contoh diatas, dengan waktu CPU=4:

```
Antrian Proses beserta Waktunya = [['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9], ['E', 1], ['D', 2], ['C', 8], ['B', 5],
['A', 16]]
Iterasi ke- 1 :
  Proses A sedang diproses, dan sisa waktu proses A = 12
  Data proses yang tersisa : [['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9], ['E', 1], ['D', 2], ['C', 8],
['B', 5]]
Iterasi ke- 2 :
  Proses B sedang diproses, dan sisa waktu proses B = 1
  Data proses yang tersisa : [['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9], ['E', 1], ['D', 2],
['C', 8]]
Iterasi ke- 3 :
  Proses C sedang diproses, dan sisa waktu proses C = 4
  Data proses yang tersisa : [['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9], ['E', 1],
['D', 2]]
Iterasi ke- 4 :
  Proses D telah selesai diproses
  Data proses yang tersisa : [['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9], ['E', 1]]
Iterasi ke- 5 :
  Proses E telah selesai diproses
  Data proses yang tersisa : [['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3], ['F', 9]]
Iterasi ke- 6 :
  Proses F sedang diproses, dan sisa waktu proses F = 5
  Data proses yang tersisa : [['F', 5], ['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9], ['G', 3]]
Iterasi ke- 7 :
  Proses G telah selesai diproses
  Data proses yang tersisa : [['F', 5], ['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4], ['H', 9]]
Iterasi ke- 8 :
  Proses H sedang diproses, dan sisa waktu proses H = 5
  Data proses yang tersisa : [['H', 5], ['F', 5], ['C', 4], ['B', 1], ['A', 12], ['J', 11], ['I', 4]]
Iterasi ke- 9 :
  Proses I telah selesai diproses
  Data proses yang tersisa : [['H', 5], ['F', 5], ['C', 4], ['B', 1], ['A', 12], ['J', 11]]
```

```
Iterasi ke- 9 :
  Proses I telah selesai diproses
  Data proses yang tersisa : [['H', 5], ['F', 5], ['C', 4], ['B', 1], ['A', 12], ['J', 11]]
Iterasi ke- 10 :
  Proses J sedang diproses, dan sisa waktu proses J = 7
  Data proses yang tersisa : [['J', 7], ['H', 5], ['F', 5], ['C', 4], ['B', 1], ['A', 12]]
Iterasi ke- 11 :
  Proses A sedang diproses, dan sisa waktu proses A = 8
  Data proses yang tersisa : [['A', 8], ['J', 7], ['H', 5], ['F', 5], ['C', 4], ['B', 1]]
Iterasi ke- 12 :
  Proses B telah selesai diproses
  Data proses yang tersisa : [['A', 8], ['J', 7], ['H', 5], ['F', 5], ['C', 4]]
Iterasi ke- 13 :
  Proses C telah selesai diproses
  Data proses yang tersisa : [['A', 8], ['J', 7], ['H', 5], ['F', 5]]
Iterasi ke- 14 :
  Proses F sedang diproses, dan sisa waktu proses F = 1
  Data proses yang tersisa : [['F', 1], ['A', 8], ['J', 7], ['H', 5]]
Iterasi ke- 15 :
  Proses H sedang diproses, dan sisa waktu proses H = 1
  Data proses yang tersisa : [['H', 1], ['F', 1], ['A', 8], ['J', 7]]
Iterasi ke- 16 :
  Proses J sedang diproses, dan sisa waktu proses J = 3
  Data proses yang tersisa : [['J', 3], ['H', 1], ['F', 1], ['A', 8]]
Iterasi ke- 17 :
  Proses A sedang diproses, dan sisa waktu proses A = 4
  Data proses yang tersisa : [['A', 4], ['J', 3], ['H', 1], ['F', 1]]
Iterasi ke- 18 :
  Proses F telah selesai diproses
  Data proses yang tersisa : [['A', 4], ['J', 3], ['H', 1]]
Iterasi ke- 19 :
  .....
```

```
Iterasi ke- 18 :
  Proses F telah selesai diproses
  Data proses yang tersisa : [['A', 4], ['J', 3], ['H', 1]]
Iterasi ke- 19 :
  Proses H telah selesai diproses
  Data proses yang tersisa : [['A', 4], ['J', 3]]
Iterasi ke- 20 :
  Proses J telah selesai diproses
  Data proses yang tersisa : [['A', 4]]
Iterasi ke- 21 :
  Proses A telah selesai diproses
  Data proses yang tersisa : []
```

### Contoh Proses Penjadwalan CPU untuk 3 Proses :

#### Input Data Proses dan Waktu Proses :

---

```
Jumlah Proses yang akan dijadwal di CPU = 3
Nama Proses ke-0 : A
Waktu proses : 5
Nama Proses ke-1 : B
Waktu proses : 9
Nama Proses ke-2 : C
Waktu proses : 2
```

#### Print data Proses :

---

```
Antrian Proses :
[['C', 2], ['B', 9], ['A', 5]]
```

#### Proses Penjadwalan CPU untuk 3 Proses seperti contoh diatas, dengan waktu CPU = 3:

---

```
waktu proses CPU = 3
Antrian Proses beserta Waktunya = [['C', 2], ['B', 9], ['A', 5]]
Iterasi ke- 1 :
    Proses A sedang diproses, dan sisa waktu proses A = 2
    Data proses yang tersisa : [['A', 2], ['C', 2], ['B', 9]]
Iterasi ke- 2 :
    Proses B sedang diproses, dan sisa waktu proses B = 6
    Data proses yang tersisa : [['B', 6], ['A', 2], ['C', 2]]
Iterasi ke- 3 :
    Proses C telah selesai diproses
    Data proses yang tersisa : [['B', 6], ['A', 2]]
Iterasi ke- 4 :
    Proses A telah selesai diproses
    Data proses yang tersisa : [['B', 6]]
Iterasi ke- 5 :
    Proses B sedang diproses, dan sisa waktu proses B = 3
    Data proses yang tersisa : [['B', 3]]
Iterasi ke- 6 :
    Proses B telah selesai diproses
    Data proses yang tersisa : []
```

---

3. Buat implementasi dari Deque yaitu Palindrom, yaitu program yang dapat melakukan pengecekan apakah suatu **kata maupun bilangan** adalah palindrom. Caranya dengan memeriksa apakah ujung yang terletak pada *front* dan ujung yang terletak pada *rear*, merupakan karakter yang sama.

```
In [19]: print(cekPalindrom('hannah'))      True
         print(cekPalindrom('surabaya'))      False
         print(cekPalindrom('abcdcba'))      True
         print(cekPalindrom('katak'))         True
         print(cekPalindrom('taat'))          True
         print(cekPalindrom('dia'))           False
```