

AULA10

ESTRUTURA DE DADOS

Fila (implementação estática)

Fila

É uma estrutura linear na qual:

Fila

É uma estrutura linear na qual:

- As **inserções** ocorrem no **final** da fila;

Fila

É uma estrutura linear na qual:

- As **inserções** ocorrem no **final** da fila;
- As **exclusões** ocorrem no **início** da fila.

Fila

É uma estrutura linear na qual:

- As **inserções** ocorrem no **final** da fila;
- As **exclusões** ocorrem no **início** da fila.
- Utiliza a mesma lógica de uma **fila de pessoas**.

Fila - implementação estática

Utilizaremos um **arranjo** de elementos de tamanho predefinido;

Fila - implementação estática

Utilizaremos um **arranjo** de elementos de tamanho predefinido;

Controlaremos a posição do elemento que está no **início** da fila.

Controlaremos o **número de elementos** da fila.

Fila (ideia geral)

Temos um **arranjo** de elementos, um campo indicando a **posição do primeiro** e um indicando o **número de elementos**. O sucessor de cada elemento está na **próxima posição do arranjo** (exceto o sucessor do último que estará na posição 0)

A	?	?	5	7	2
inicio	2				
nroElem	3				

Fila (ideia geral)

Temos um **arranjo** de elementos, um campo indicando a **posição do primeiro** e um indicando o **número de elementos**

O sucessor de cada elemento está na **próxima posição do arranjo** (exceto o sucessor do último que estará na posição 0)

Como inserimos o **elemento 8**?

A	?	?	5	7	2
inicio	2				
nroElem	3				

Fila (ideia geral)

Temos um **arranjo** de elementos, um campo indicando a **posição do primeiro** e um indicando o **número de elementos**

O sucessor de cada elemento está na **próxima posição do arranjo** (exceto o sucessor do último que estará na posição 0)

Como inserimos o **elemento 8**?

A	8	?	5	7	2
inicio	2				
nroElem	4				

Fila (ideia geral)

Temos um **arranjo** de elementos, um campo indicando a **posição do primeiro** e um indicando o **número de elementos**

O sucessor de cada elemento está na **próxima posição do arranjo** (exceto o sucessor do último que estará na posição 0)

Como inserimos o **elemento 8**?

Como **excluimos um elemento**?

A	8	?	5	7	2
inicio	2				
nroElem	4				

Fila (ideia geral)

Temos um **arranjo** de elementos, um campo indicando a **posição do primeiro** e um indicando o **número de elementos**

O sucessor de cada elemento está na **próxima posição do arranjo** (exceto o sucessor do último que estará na posição 0)

Como inserimos o **elemento 8**?

Como **excluimos um elemento**?

A	8	?	5	7	2
inicio	3				
nroElem	3				

Modelagem

```
#include <stdio.h>
```

```
#define MAX 50
```

```
typedef int bool;
```

```
typedef int TIPOCHAVE;
```

```
typedef struct {  
    TIPOCHAVE chave;  
} REGISTRO;
```

```
typedef struct {  
    REGISTRO A[MAX];  
    int inicio;  
    int nroElem;  
} FILA;
```

Modelagem

```
#include <stdio.h>
```

```
#define MAX 50
```

```
typedef int bool;
```

```
typedef int TIPOCHAVE;
```

```
typedef struct {  
    TIPOCHAVE chave;  
} REGISTRO;
```

```
typedef struct {  
    REGISTRO A[MAX];  
    int inicio;  
    int nroElem;  
} FILA;
```

Funções de gerenciamento

Implementaremos funções para:

- Inicializar a estrutura

- Retornar a quantidade de elementos válidos

- Exibir os elementos da estrutura

- Inserir elementos na estrutura (no fim)

- Excluir elementos da estrutura (no início)

- Reinicializar a estrutura

Inicialização

Para inicializarmos nossa fila (implementação estática), **precisamos:**

Inicialização

Para inicializarmos nossa fila (implementação estática), **precisamos**:

- Acertar o valor do campo **nroElem** (para indicar que não há nenhum elemento válido);
- Acertar o valor do campo **inicio** (índice do primeiro elemento válido)?

Inicialização

```
void inicializarFila(FILA* f) {  
    f->inicio=0;  
    f->nroElem=0;  
}
```

2010

A	?	?	?	?	?
inicio	?				
nroElem	?				

Inicialização

```
void inicializarFila(FILA* f) {  
    f->inicio=0;  
    f->nroElem=0;  
}
```

f 2010 2010

A	?	?	?	?	?
inicio	?				
nroElem	?				

Inicialização

```
void inicializarFila(FILA* f) {  
    f->inicio=0;  
    f->nroElem=0;  
}
```

f 2010 2010

A	?	?	?	?	?
inicio	0				
nroElem	0				

Retornar número de elementos

Retornar número de elementos

Basta retornarmos o valor do campo *nroElem*.

```
int tamanhoFila(FILA* f) {  
    return f->nroElem;  
}
```

Exibição/Impressão

Para exibir os elementos da estrutura precisaremos iterar pelos **elementos** válidos.

Exibição/Impressão

Para exibir os elementos da estrutura precisaremos iterar pelos **elementos** válidos.

Atenção:

Há *nroElem* elementos válidos e o primeiro está na posição *inicio* do arranjo

Após o elemento da última posição do arranjo (posição *MAX-1*) está o elemento da *posição 0* (trataremos o arranjo como se fosse **circular**)

Exibição/Impressão

```
void exhibirFila(FILA* f) {  
    printf("Fila: \" ");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\n");  
}
```

Exibição/Impressão

```
void exhibirFila(FILA* f) {  
    printf("Fila: \" ");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\\n\\n");  
}
```

Exibição/Impressão

```
void exhibirFila(FILA* f) {  
    printf("Fila: \" \");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\n");  
}
```

Exibição/Impressão

```
void exhibirFila(FILA* f) {  
    printf("Fila: \" \");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\n");  
}
```

Exibição/Impressão

```
void exibirFila(FILA* f) {  
    printf("Fila: \" ");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\\n\\n");  
}
```

A	8	?	5	7	2
inicio	2				
nroElem	4				

Exibição/Impressão

```
void exibirFila(FILA* f) {  
    printf("Fila: \" ");  
    int i = f->inicio;  
    int temp;  
    for (temp = 0; temp < f->nroElem; temp++) {  
        printf("%i ", f->A[i].chave);  
        i = (i + 1) % MAX;  
    }  
    printf("\\n\\n");  
}
```

A	8	?	5	7	2
inicio	2				
nroElem	4				

\$ Fila: " 5 7 2 8 "

Inserção de um elemento

O usuário passa como parâmetro um registro a ser inserido no final da fila

Se a fila não estiver **cheia**, **precisamos**:

Inserção de um elemento

O usuário passa como parâmetro um registro a ser inserido no final da fila

Se a fila não estiver **cheia**, **precisamos**:

Identificar a **posição** no arranjo na qual o registro será inserido e inseri-lo lá;

Inserção de um elemento

O usuário passa como parâmetro um registro a ser inserido no final da fila

Se a fila não estiver **cheia**, **precisamos**:

Identificar a **posição** no arranjo na qual o registro será inserido e inseri-lo lá;

Alterar o valor campo ***nroElem***.

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8

2010

A	?	?	5	7	2
inicio	2				
nroElem	3				

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8

2010

A	?	?	5	7	2
inicio	2				
nroElem	3				

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8
posicao	0

2010

A	?	?	5	7	2
inicio	2				
nroElem	3				

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8
posicao	0

2010

A	8	?	5	7	2
inicio	2				
nroElem	3				

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8
posicao	0

2010

A	8	?	5	7	2
inicio	2				
nroElem	4				

Inserção de um elemento

```
bool inserirElementoFila(FILA* f, REGISTRO reg) {  
    if (f->nroElem >= MAX) return false;  
    int posicao = (f->inicio + f->nroElem) % MAX;  
    f->A[posicao] = reg;  
    f->nroElem++;  
    return true;  
}
```

f	2010
reg	8
posicao	0

2010

A	8	?	5	7	2
inicio	2				
nroElem	4				

Exclusão de um elemento

O usuário solicita a exclusão do elemento do **início da fila**. Se a fila **não estiver vazia**:

Exclusão de um elemento

O usuário solicita a exclusão do elemento do **início da fila**. Se a fila **não estiver vazia**:

- Iremos **copiar esse elemento para um local indicado pelo usuário**;
- Acertar o valor do campo ***nroElem***;
- Acertar o valor do campo ***inicio***.

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
  
  
  
  
  
  
  
  
}
```

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
    if (f->nroElem==0) return false;  
  
    }  
}
```

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
    if (f->nroElem==0) return false;  
    *reg = f->A[f->inicio];  
  
}
```

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
    if (f->nroElem==0) return false;  
    *reg = f->A[f->inicio];  
    f->inicio = (f->inicio+1) % MAX;  
  
}
```

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
    if (f->nroElem==0) return false;  
    *reg = f->A[f->inicio];  
    f->inicio = (f->inicio+1) % MAX;  
    f->nroElem--;  
  
}
```

Exclusão de um elemento

```
bool excluirElementoFila(FILA* f, REGISTRO* reg) {  
    if (f->nroElem==0) return false;  
    *reg = f->A[f->inicio];  
    f->inicio = (f->inicio+1) % MAX;  
    f->nroElem--;  
    return true;  
}
```

Reinicialização da fila estática

Para reinicializar esta estrutura basta chamarmos a **função de inicialização** ou executarmos os mesmos comandos executados lá.

Reinicialização da fila estática

```
void reinicializarFila(FILA* f) {  
    inicializarFila(f);  
}
```

```
void reinicializarFila2(FILA* f) {  
    f->inicio=0;  
    f->nroElem=0;  
}
```

AULA10

ESTRUTURA DE DADOS

Fila (implementação estática)