

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação `cria_fila()` que compõem o TAD `FILA_SEQ`.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILA_SEQ;
void cria_fila (FILA_SEQ *);
int eh_vazia (FILA_SEQ *);
int tam (FILA_SEQ *);
void ins (FILA_SEQ *, int);
int cons (FILA_SEQ *);
void ret (FILA_SEQ *);
int cons_ret (FILA_SEQ *);
```

```
void cria_fila (FILASEQ *f)  
{  
    f->N = f->INICIO = 0;  
    f->FIM = -1;  
}
```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação `eh_vazia()` que compõem o

TAD `FILA_SEQ`.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILAS_SEQ;
void cria_fila (FILAS_SEQ *);
int eh_vazia (FILAS_SEQ *);
int tam (FILAS_SEQ *);
void ins (FILAS_SEQ *, int);
int cons (FILAS_SEQ *);
void ret (FILAS_SEQ *);
int cons_ret (FILAS_SEQ *);
```

```
int eh_vazia (FILASEQ *f)
{
    return (!f->N);
}
```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação tam() que compõem o TAD

FILA_SEQ.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILAS_SEQ;
void cria_fila (FILAS_SEQ *);
int eh_vazia (FILAS_SEQ *);
int tam (FILAS_SEQ *);
void ins (FILAS_SEQ *, int);
int cons (FILAS_SEQ *);
void ret (FILAS_SEQ *);
int cons_ret (FILAS_SEQ *);
```

```
int tam (FILA_SEQ *f)
{
    return (f->N);
}
```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação `ins()` que compõem o TAD

`FILA_SEQ.`

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILA_SEQ;
void cria_fila (FILA_SEQ *);
int eh_vazia (FILA_SEQ *);
int tam (FILA_SEQ *);
void ins (FILA_SEQ *, int);
int cons (FILA_SEQ *);
void ret (FILA_SEQ *);
int cons_ret (FILA_SEQ *);
```

```
void ins (FILASEQ *f, int v)  
{  
    if (f->N == MAX)  
    {  
        printf ("\nERRO! Estouro na fila.\n");  
        exit (1);  
    }  
    f->FIM = ((f->FIM)+1) % MAX;  
    f->val[f->FIM]=v;  
    f->N++;  
}
```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação `cons()` que compõem o TAD

`FILA_SEQ`.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILA_SEQ;
void cria_fila (FILA_SEQ *);
int eh_vazia (FILA_SEQ *);
int tam (FILA_SEQ *);
void ins (FILA_SEQ *, int);
int cons (FILA_SEQ *);
void ret (FILA_SEQ *);
int cons_ret (FILA_SEQ *);
```

```
int cons (FILASEQ *f)  
{  
    if (eh_vazia(f))  
    {  
        printf ("\nERRO! Consulta na fila  
vazia.\n");  
        exit (2);  
    }  
    else  
        return (f->val[f->INICIO]);  
}
```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação ret() que compõem o TAD

FILA_SEQ.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILA_SEQ;
void cria_fila (FILA_SEQ *);
int eh_vazia (FILA_SEQ *);
int tam (FILA_SEQ *);
void ins (FILA_SEQ *, int);
int cons (FILA_SEQ *);
void ret (FILA_SEQ *);
int cons_ret (FILA_SEQ *);
```

```

void ret (FILASEQ *f)
{
    if (eh_vazia(f))
    {
        printf ("\nERRO! Retirada na fila
vazia.\n");
        exit (3);
    }
    else
    {
        f->INICIO= (f->INICIO+1) % MAX;
        f->N--;
    }
}

```

Fila - Alocação Sequencial

Com base no que foi visto implemente a operação `cons_ret()` que compõem o TAD `FILA_SEQ`.

```
typedef struct
{
    int N;
    int INICIO;
    int FIM;
    int val[MAX];
}FILA_SEQ;
void cria_fila (FILA_SEQ *);
int eh_vazia (FILA_SEQ *);
int tam (FILA_SEQ *);
void ins (FILA_SEQ *, int);
int cons (FILA_SEQ *);
void ret (FILA_SEQ *);
int cons_ret (FILA_SEQ *);
```

```

int cons_ret (FILA_SEQ *f)
{
    if (eh_vazia(f))
    {
        printf ("\nERRO! Consulta e retirada na
fila vazia.\n");
        exit (4);
    }
    else
    {
        int v=f->val[f->INICIO];
        f->INICIO= (f->INICIO+1) % MAX;
        f->N--;
        return (v);
    }
}

```

Alocação Sequencial - Exercício

Implemente, no TAD `FILA_SEQ`, utilizando recursividade, a seguinte operação:

```
void gera_fila (FILA_SEQ *f, int m, int n);
```

a qual utilizando-se das operações do TAD `FILA_SEQ` produz uma fila de inteiros correspondente a `[m..n]`.