



## ***Prova 2 – Fila e Pilha***

### **Instruções:**

1. Para resolver as questões da prova, considere a existência das bibliotecas “*pilha.h*” e “*fila.h*”, conforme implementações e discussões realizadas em aula.
2. A prova pode ser resolvida com consulta ao material impresso ou digital **off-line** (ou seja, não será permitido o acesso à internet durante a prova).
3. **Provas e/ou questões idênticas (ou com indícios de cópia) terão a nota zerada.**
4. Ao término da prova, compacte a implementação de cada questão em um único arquivo (*com o seu nome completo*) e submeta no moodle na atividade denominada “**Prova 2 – Fila e Pilha**”.

**Questão 1 – (2,5 pontos)** Escreva uma função denominada NGE (*Next Greater Element*) que recebe como parâmetro uma **fila** e retorna uma **lista encadeada** com o mapeamento dos elementos da fila com seu respectivo NGE.

O NGE de um dado elemento **x** é o primeiro elemento maior que está na sequência da fila. Quando um elemento não tiver um NGE correspondente, considere o NGE sendo -1.

### **Exemplos:**

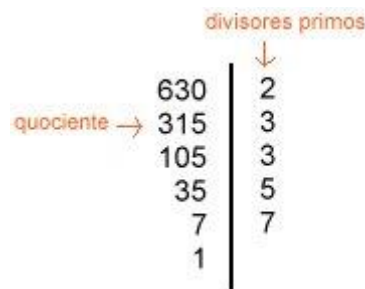
- a) Para qualquer fila, o NGE do último elemento da fila é sempre -1.
- b) Para qualquer fila cujos elementos estiverem ordenados decrescentemente, o NGE de todos os elementos são -1.
- c) Para uma fila (onde o elemento da esquerda é o primeiro) formada pela sequência de elementos [4, 5, 2, 25], o mapeamento dos elementos com seu respectivo NGE são os seguintes:

Elemento	NGE
4	5
5	25
2	25
25	-1

**struct DLista\* NGE (struct DFila \*fila, FuncaoComparacao fc)**

**Questão 2 - (2,5 pontos).** Para um dado número inteiro  $N > 1$ , o **menor** inteiro  $D > 1$  que divide  $N$  é chamado de **fator primo**. É possível determinar a fatoraçoão prima de  $N$  achando-se o fator primo  $D$  e substituindo  $N$  pelo quociente  $N/D$ , repetindo essa operação até que  $N$  seja igual a 1.

A figura a seguir ilustra a fatoraçoão prima do número 630.



Implemente uma função que compute a fatoraçoão prima de um número  $N$  qualquer e retorne uma fila contendo a sequência de fatores primos do referido número  $N$ , conforme a assinatura a seguir:

```
struct DFila* fatoracaoPrima (int n)
```

**Questão 3 – (2,5 pontos)** Escreva uma função “troca” que recebe como parâmetros uma pilha e um número inteiro  $p$  e troca o elemento do topo com o elemento da posição  $p$ .

```
void troca (struct DPilha *pilha, int posicao)
```

**Questão 4 - (2,5 pontos)** Considere uma sequência de E's e D's que representam, respectivamente, as ações de empilhar e desempilhar elementos em/de uma pilha.

Escreva uma função que recebe uma sequência qualquer de E's e D's representando uma sequência bem formada de ações de empilhamento e desempilhamento, retornando 'S' quando a sequência for bem-formada e 'N' quando for mal-formada.

Exemplos:

- **EEEDD**: bem-formada, pois nunca realizada o desempilhamento com a pilha vazia.
- **EDEDEEDDDE**: mal-formada, pois quando for executar a última ação de desempilhamento (indicada em vermelho), a pilha já estará vazia.

```
char bemFormada (char sequencia[])
```