sqtpm
[183012]

voltar

Trabalho: 04-din-circ
Linguagens: C
Arquivos-fonte a enviar: 3.
Data de abertura: 2016/09/12 14:00:00
Data limite para envio: 2016/09/19 12:00:00
Número máximo de envios: 25
Casos-de-teste abertos: casos-de-teste.tgz

Seqüência circular em vetor dinâmico

Vamos dizer que um conjunto de registros forma uma *seqüência* se eles são consecutivos e se a ordem relativa entre eles é importante.

Neste trabalho você deve implementar um vetor dinâmico para armazenar uma seqüência. O vetor dinâmico deve permitir inserção no início e no final da seqüência, remoção do primeiro e do último elementos e recuperação do primeiro e do último elementos. A seqüência deve ser circular, isto é, o primeiro elemento dela deve poder estar em qualquer posição do vetor. Dessa forma, qualquer uma dessas operações pode ser realizada em tempo constante.

A seqüência circular de chaves inteiras [2,9,5,7] pode estar armazenada de várias formas em um vetor de tamanho 8, por exemplo:

```
2 9 5 7 _ _ _ _
_ _ 2 9 5 7 _ _
_ _ _ _ 2 9 5 7
5 7 _ _ _ _ 2 9
9 5 7 _ _ _ _ 2
```

Em todas elas, o primeiro elemento da seqüência é o 2 e o último é o 7.

Durante o processamento, o número de posições vazias do vetor não deve exceder 3n, onde n é o número de posições ocupadas. A política de crescimento do vetor deve ser dobrar quando estiver cheio e reduzir à metade quando estiver 1/4 ocupado. Antes de terminar o programa deve liberar a memória ocupada pelo vetor dinâmico.

O vetor dinâmico deve ser implementado em um par de arquivos c e h separados. O arquivo h deve conter as declarações do tipo de dados e das operações sobre o vetor dinâmico. O arquivo c deve definir as funções. O programa principal deve estar em um terceiro arquivo c.

Neste trabalho, o vetor dinâmico deve armazenar seqüências de cadeias de caracteres. A memória usada por cada cadeia de caracteres deve ser a menor possível.

Entrada

A entrada é composta por uma sucessão de comandos, um por linha. Os possíveis comandos estão descritos abaixo.

- `unshift cadeia`

  Insere uma cadeia no início da seqüência. A cadeia é formada pelos caracteres depois de "unshift " e antes do fim-de-linha, descartando os espaços no início e no fim-de-linha. A cadeia pode ser vazia.

- `shift`

  Remove a cadeia no início da seqüência. Se a seqüência estiver vazia, não faz nada.

- `print-first`

  Imprime a cadeia no início da seqüência. Se a seqüência estiver vazia, não faz nada.

- `push cadeia`

  Insere uma cadeia no fim da seqüência. A cadeia é formada pelos caracteres depois de "push " e antes do fim-de-linha, descartando os espaços no início e no fim-de-linha. A cadeia pode ser vazia.

- `pop`

  Remove a cadeia no fim da seqüência. Se a seqüência estiver vazia, não faz nada.

- `print-last`

  Imprime a cadeia no fim da seqüência. Se a seqüência estiver vazia, não faz nada.

- `is-empty`

  Imprime yep se a seqüência estiver vazia e nope se não estiver.

- `exit`

  Termina o programa.

Saída

A saída deve conter as linhas geradas pelos comandos print-first e print-last.

Exemplo:

Entrada:

```
is-empty
pop
print-last
push   mas um dia
push desses
push eu
print-first
print-last
shift
print-first
is-empty
exit
```

Saída:

```
yep
mas um dia
eu
desses
nope
```
