

Dredd - Juiz Online

Principal

Perfil

Minhas Provas

Sair

Minutos
Restantes:
41016

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Total: 0

Exercícios sobre Pilhas e Filas

Prova Aberta Até: 10/10/2018 11:08:06

Número Máximo de Tentativas: 6

Atenuação da Nota por Tentativa: 2%

Instruções para a prova: A prova é individual. Desligue seu celular. Não converse com os colegas. Não fique olhando para a tela dos colegas.

Questão 1: Pilha com armazenamento de dados em arranjos

Implemente uma pilha, utilizando armazenamento dos dados em arranjos. Segue esboço da implementação:

```
typedef int Dado; // para facilitar a troca de int para outro tipo

// pilha implementada em arranjo
class pilhav {
private:
    int capacidade;
    Dado *dados;
    int tamanho;
    int posTopo;
public:
    pilhav(int cap = 100);
    ~pilhav();
    void empilha(Dado valor);
    Dado desempilha();
    Dado espia(); // acessa elemento do topo, mas não retira
    void depura(); // imprime os dados do vetor da pilha sem removê-los (desempilhar)
    void info(); // imprime informações da pilha (tamanho e posição do topo)
};
```

Não é necessário tratar problemas do tipo retirada de elemento de pilha vazia ou empilhamento em pilha cheia. A aplicação principal deverá criar uma pilha com capacidade para 20 elementos e realizar as seguintes operações na ordem apresentada:

- Empilhar cinco elementos
- Desempilhar três elementos (imprimindo-os)
- Empilhar mais quatro elementos
- Desempilhar três elementos (imprimindo-os)
- Imprime o topo da pilha, sem desempilhar (`espia()`)
- Imprime os elementos restantes da pilha (`depura()`)
- Imprime tamanho e posição do topo (`info()`)

Entradas:

1. Cinco valores a serem empilhados
2. Quatro valores a serem empilhados

Saídas:

1. Três elementos desempilhados
2. Três elementos desempilhados
3. Topo da pilha
4. Elementos restantes da pilha
5. Tamanho e posição do topo

Exemplo de Entrada:

```
8 9 1 3 5
12 23 0 3
```

Exemplo de Saída:

```
5 3 1
3 0 23
```

12
8 9 12
3 2

Minutos
Restantes:
41016

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Total: 0

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Choose File No file chosen

Enviar Resposta

Questão 2: Pilha com armazenamento dinâmico

Usando o código com as especificações de métodos e atributos mais um programa que usa a classe Pilha, crie uma classe pilha com armazenamento dinâmico.

O método Desempilhar deve escrever alguma mensagem de erro na saída de erros quando não for possível desempilhar.

Entradas:

O programa passado aceita os seguintes comandos:

- i seguido de um valor inteiro: para inserir um item na pilha
- r: para remover um item da pilha
- l: para limpar (remover todos) os itens da pilha
- t: para consultar o tamanho da pilha
- e: para espiar o topo da pilha (escreve o topo sem removê-lo)
- f: para finalizar a execução do programa

Saídas:

Todas as saídas de comandos já estão implementadas na função principal. Ao terminar a execução do programa, todos os itens da pilha são desempilhados e escritos, mesmo assim, o destrutor deve desalocar memória de todos os itens da pilha.

Exemplo de Entrada e Saída juntas:

```
i 2
i 4
i 6
r
6
t
2
l
i 8
i 10
e
10
f
10 8
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Choose File No file chosen

Enviar Resposta

Questão 3: Fila com armazenamento dinâmico

Usando o código com as especificações de métodos e atributos mais um programa que usa a classe Fila, crie uma classe fila com armazenamento dinâmico.

Nas situações em algum método não puder ser executado, ele deve só escrever alguma mensagem de erro na saída de erros.

Minutos
Restantes:
41016

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Total: 0

Entradas:

O programa passado aceita os seguintes comandos:

- i seguido de um valor inteiro: para inserir um item na fila
- r: para remover um item da fila
- l: para limpar (remover todos) os itens da fila
- t: para consultar o tamanho da fila
- e: para espiar o próximo da fila (escreve o primeiro sem removê-lo)
- f: para finalizar a execução do programa

Saídas:

Todas as saídas de comandos já estão implementadas na função principal. Ao terminar a execução do programa, todos os itens da fila são removidos e escritos, mesmo assim, o destrutor deve desalocar memória de todos os itens da fila.

Exemplo de Entrada e Saída juntas:

```
i 2
i 4
i 6
r
2
t
2
l
i 8
i 10
e
8
f
8 10
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Choose File No file chosen

Enviar Resposta

Questão 4: Fila - Implementação em vetor

Filas podem ser implementadas com armazenamento de dados em vetor, se a capacidade fixa não for um problema. Nesse caso, é importante controlar os índices para "dar a volta" no vetor sempre que for necessário.

O [programa fornecido](#) contém os métodos e atributos para uma implementação assim, juntamente com um programa para testar a classe fila. Complete a implementação, fazendo com que a inserção (método Enfileirar) sempre use o início do vetor (posição zero) quando a fila estiver vazia.

Entradas:

Inicialmente, o programa lê um número inteiro que é capacidade da fila a ser criada.

Depois, o programa lê códigos de comandos a executar. Sempre um caractere identificando o comando seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- t - para terminar a execução do programa
- e - para enfileirar um valor - seguido do valor (número inteiro)
- d - para desenfileirar um valor
- p - para escrever o próximo valor na fila (sem desenfileirar)
- i - para escrever as informações da fila (atributos e conteúdo)

Saídas:

Os comandos d, p e i são os únicos que produzem saída de dados. Cada um pode ser observado no exemplo.

Exemplo de Entrada e Saída juntos:

```
3
e 1
e 2
e 3
i
tamanho=3 capacidade=3 inicio=0 fim=2
```

Minutos
Restantes:
41016

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Total: 0

```
1 2 3
d
1
e 4
i
tamanho=3 capacidade=3 inicio=1 fim=0
4 2 3
d
2
d
3
i
tamanho=1 capacidade=3 inicio=0 fim=0
4 2 3
t
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Choose File No file chosen

Enviar Resposta

Questão 5: Deque - controle de undo

Vários programas tem um comando para desfazer a operação anterior, conhecido como *undo*. Usar este comando seguidamente faz com que a última ação seja desfeita, depois a penúltima e assim por diante.

Para não gastar muita memória é comum que haja um limite de ações disponíveis para o *undo*. Por exemplo, pode ser possível desfazer apenas as últimas 20 operações realizadas.

Este tipo de funcionalidade é mais fácil de controlar usando uma estrutura do tipo *deque* (*double ended queue*). Use alguma implementação de fila ou de lista para criar uma classe Deque. Depois, faça um programa para guardar e retirar linhas de texto, que representam comandos.

Entradas:

Inicialmente seu programa deve ler um número natural que representa a quantidade máxima de instruções que a estrutura de dados vai guardar. Depois várias linhas de texto serão lidas. Sempre que a linha for "undo", o programa deve dizer que desfez a última instrução. Se a instrução não for "undo", o programa deve guardar a linha na lista de coisas a desfazer. Se a capacidade máxima for atingida, o programa deve "esquecer" a instrução mais antiga antes de inserir a nova instrução na estrutura. O comando especial "sair" faz com que a execução termine.

1. Capacidade da fila de comandos;
2. comandos, cada um numa linha.

Saídas:

O comando "undo" deve produzir a saída: desfazer: comando que foi desfeito. Toda vez que a capacidade máxima for atingida, o programa deve produzir a saída: esqueci: comando que foi removido. No caso de tentarem desfazer ação que não existe, o programa deve escrever ERRO (letras maiúsculas) na saída padrão.

Exemplo de Entrada:

```
3
comando um
comando dois
undo
comando três
comando quatro
comando cinco
sair
```

Exemplo de Saída:

```
desfazer: comando dois
esqueci: comando um
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

No file chosen

Minutos
Restantes:
41016

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Q3: ?
Q4: ?
Q5: ?
Q6: ?
Total: 0

Questão 6: Filas - Mesclar duas filas

Implemente, a partir [deste código](#), o construtor de cópia, o operador de atribuição (=) e a sobrecarga do operador de soma (+) em uma fila. A soma deverá ter o significado de mesclar duas filas de números inteiros previamente ordenadas e retornar uma nova fila totalmente ordenada. Não é necessário verificar se as filas estavam ordenadas, simplesmente suponha que estavam.

O programa tem duas filas (A e B) e pode inserir elementos em qualquer uma, copiar os valores de qualquer uma para a outra, escrever os dados de qualquer uma ou mesclar A com B numa fila temporária.

Entradas:

Uma sequência de comandos para o programa executar, a saber:

- i: seguido de um identificador de fila (a ou b), seguida de um número inteiro, insere a palavra na fila
- e: seguido de um identificador de fila (a ou b), escreve o conteúdo da fila
- a=b: copia a fila b para a fila a (note a ausência de espaços no comando)
- b=a: copia a fila a para a fila b (note a ausência de espaços no comando)
- s: soma (mescla) a fila a com a fila b numa fila temporária e escreve o seu conteúdo
- t: termina a execução do programa

Saídas:

O comando e escreve o conteúdo de uma das listas. O comando s escreve o resultado de mesclar a com b.

Exemplo de Entrada:

```
i a 1
i a 3
i a 5
i b 2
i b 4
i b 6
e a
e b
s
a=b
e a
e b
t
```

Exemplo de Saída:

```
1 3 5
2 4 6
1 2 3 4 5 6
2 4 6
2 4 6
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

 No file chosen

Desenvolvido por Bruno Schneider a partir
do programa original (Algod) de Renato R.
R. de Oliveira.

