

Pilhas e filas

Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, **copie e cole o seu código-fonte com a resposta aqui mesmo neste documento**, dentro dos espaços indicados para isso e **preservando a indentação do código**. **Depois que terminar sua avaliação, não se esqueça de entregar sua atividade!** Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

Para resolver esta atividade, [clique aqui para baixar](#) o projeto que contém a implementação do TAD `FilaEncad` de números inteiros. Na sua solução para a questão abaixo, [você pode utilizar/chamar](#) qualquer uma das operações que estejam disponíveis no projeto (exatamente do jeito que ele se encontra no site da disciplina). Outras operações que você eventualmente deseje utilizar devem ser implementadas e incluídas na sua resposta neste documento.

Suponha uma fila de pessoas esperando para comprar sorvete que custa 5 dinheiros. Inicialmente, o caixa tem zero dinheiros para oferecer de troco e cada elemento da fila representa a nota que a pessoa vai utilizar para o pagamento. Escreva uma função no programa principal `bool verifica_troco(FilaEncad *f)` que recebe a fila como parâmetro e verifica se a sorveteria tem troco para todos que estão na fila.

Dica: para simplificar, considere para troco apenas as notas de 5 e de 10 e considere que 20 será o valor máximo usado para pagamento. Considere também que a verificação do troco será feita somente no momento em que a pessoa estiver sendo atendida, ou seja, a pessoa não poderá aguardar pelo troco de outra que está atrás dela na fila.

Exemplos:

Para uma fila original contendo: 5 5 5 10 20

A função deve retornar: **true**

Para uma fila contendo: 5 10 10 20

A função deve retornar: **false**

Para uma fila contendo: 5 10 10 5 20

e retornar: **false**

Inserir seu código aqui:

```
bool FilaEncad::verifica_troco(FilaEncad *f)
{
    No* p;
    int valAux = 5;
    int cont = 0;
    if(f->getInicio() > 5)
```

```
{  
    return false;  
}  
for(p = f->inicio;p!=NULL;p = p->getProx())  
{  
    if(p!=f->getInicio())  
    {  
        if(p->getInfo() > 5)  
        {  
            if(p->getInfo() - valAux <= 5)  
            {  
                valAux=5;  
            }  
            else  
            {  
                return false;  
            }  
        }  
        else  
        {  
            valAux+=5;  
        }  
    }  
    cont++;  
}  
return true;  
}
```