

Laboratório de Programação C++

Estrutura de dados: Pilha

Objetivo:

Representação e operação de uma estrutura de dados que corresponde a uma lista sequencial com propriedade LIFO (“Last in First Out”), usando alocação dinâmica de memória.

Considerações:

Pilha (stack) é uma lista na qual os elementos podem ser acessados apenas na ordem último a entrar, primeiro a sair (LIFO). As pilhas suportam somente duas operações básicas: colocar (push) e retirar (pop). Na primeira, um novo elemento é colocado no topo da pilha, caso a pilha não esteja cheia; na segunda, um elemento do topo da pilha é recuperado, caso a pilha não esteja vazia. A pilha pode ser implementada com alocação dinâmica de memória usando um ponteiro **p** para acessar o seu topo. Na operação **push**, um elemento é colocado no topo da pilha, em seguida o topo é incrementado. Na operação **pop**, o topo é decrementado e o elemento do topo é retirado. Implemente e teste a operação desta pilha com a seguinte classe:

```
class Pilha {
    int tamanho;        // número máximo de elementos da pilha
    int *buffer;        // ponteiro para a área de memória da pilha
    int *topo           // ponteiro para o topo da pilha
public:
    Pilha(int t);        // construtor para criar uma pilha com tamanho t
    ~Pilha();            // destrutor
    void push(int x);    // coloca um elemento no topo da pilha
    int pop();           // retira um elemento do topo da pilha
    bool Pcheia();       // verifica se pilha está cheia
    bool Pvazia();       // verifica se pilha está vazia
    void print() ;       // mostra a pilha
};
```

Considere:

- Tamanho máximo da pilha igual a 100. Se o tamanho solicitado for maior que 100, então a pilha de tamanho máximo é criada. Se o tamanho da pilha solicitada for zero ou menos, uma pilha de tamanho 1 é criada. O tamanho da pilha é armazenado no campo `tamanho` pelo construtor
- A tentativa de colocar um elemento na pilha cheia gera um erro de **stack overflow**
- A tentativa de retirar um elemento da pilha vazia gera um erro de **stack underflow**
- Os erros devem ser previstos