

Lista de exercícios 2

1. Considerando a entrada e a sequência de operações (inserir e retirar) em uma pilha, mostre a saída resultante.

a)

Entrada: I E N S R E

Sequência: I R I I R I R I I R R R

Saída:

5	
4	
3	
2	
1	
0	

b)

Entrada: A, B, C, D, E, F

Sequência: I I R I I R I R R I R R

Saída:

5	
4	
3	
2	
1	
0	

2. Implemente um programa que manipule uma pilha. O programa deve ser desenvolvido com as seguintes regras:
- Criar um menu com as opções:
 - Criar pilha – O usuário deve informar o tamanho da pilha; caso a pilha já tenha sido criada anteriormente, antes de criar uma pilha nova, a pilha antiga deve ser removida e a memória deve ser desalocada.
 - Inserir – O usuário deve informar um valor a ser inserido; caso a pilha esteja cheia, uma mensagem deve ser exibida.
 - Remover – Remover um elemento da pilha e mostrar o elemento removido; caso a pilha esteja vazia, uma mensagem deve ser exibida.
 - Consultar – Verificar se a pilha contém um determinado valor informado pelo usuário.
 - Mostrar – Mostrar a pilha.
 - Sair – Sair do programa.
 - Não permitir a inclusão de valores duplicados;
 - Imprimir uma mensagem de erro caso a pilha não tenha sido criada e o usuário tente executar uma das seguintes operações: inserir, remover, consultar ou mostrar.
3. Faça um programa que cadastre em uma estrutura do tipo pilha vários números, sendo no máximo 6, ao remover um número desta estrutura o mesmo deve ser empilhado em outra pilha, conforme o critério: se o número for par na pilha dos pares, se for ímpar na pilha dos ímpares. No menu de opções deve ter uma alternativa para zerar as três pilhas, mostrar pilha inicial, a dos pares e a dos ímpares.
4. Crie uma pilha que permita armazenar dados do tipo string. Para isso, deve-se modificar a estrutura da Pilha, “int *dados;” para “string *dados;”. Além disso, as demais funções que recebem o valor do tipo inteiro devem ser alteradas para o tipo string. Exemplos:

```
void inicializaP(Pilha *p, int tam)
{
    .
    .
    .
    p->dados = new string[tam];///aloca memória dinamicamente
}
bool empilhaP(Pilha *p, string valor)
string desempilhaP(Pilha *p)
string espiaP(Pilha *p)
bool buscaP(Pilha *p, string valor)
```

5. Faça um programa para ler uma string e após empilhar os caracteres que são letras em uma pilha P1 e empilhar os que são dígitos em outra pilha P2. Ao final, desempilhe os dados de P2 e em seguida, os de P1, imprimindo-os na tela.

Dicas:

- Use as funções isdigit e isalpha da biblioteca “#include <cctype>”
 - A pilha P1 deve armazenar letras. Dessa forma, altere o arquivo “pilha.hpp” para atender esta necessidade.
6. Faça um programa que leia uma string e verifica se o texto é um palíndromo, ou seja, se a string é escrita da mesma maneira de frente para trás e de trás para frente. Ignore espaços e pontos. Exemplos: Osso, Radar, Ovo, Arara

Observação: Use uma pilha para realizar a verificação do palíndromo.

7. Empregando pilha, crie uma função para verificar se uma expressão matemática composta apenas por parênteses está ou não *balanceada*.

- Exemplo de expressões balanceadas: " $(A+B)$ ", " $((A+B) + (C+D))$ "
- Exemplo de expressões não balanceadas: " $((A+B)$ ", " $)A+B($ ", " $(A+B)) - (C + D)$ "

8. Desenvolva as seguintes funções:

- Função para testar se duas pilhas P1 e P2 são iguais.

Observação: Duas pilhas são iguais se elas possuem os mesmos elementos e na mesma ordem.

- Função para retornar o número de elementos da pilha que possuem valor par.

Observação: Use uma pilha auxiliar e as operações de empilha/desempilha para implementar as funções, ou seja, não é permitido acessar o vetor da pilha diretamente (pilha->dados[i]).

9. Faça um programa que usa uma pilha para armazenar o código das cidades que fazem parte de um caminho.

- Inicialmente, o usuário deve informar o número de cidades;
- Depois, o usuário deve informar o código de cada cidade e incluir na pilha;
- Criar uma funcionalidade de busca de um código na pilha: para fazer a busca deve usar uma pilha auxiliar; a cada item verificado na pilha principal que for diferente do código procurado, deve incluir na pilha auxiliar; após encontrar o valor, deve retornar os valores da pilha auxiliar para a pilha principal.

10. Resolva a questão abaixo:

A pilha é uma estrutura de dados que permite a inserção/remoção de itens dinamicamente seguindo a norma de último a entrar, primeiro a sair. Suponha que para uma estrutura de dados, tipo pilha, são definidos os comandos:

- PUSH (p, n): Empilha um número "n" em uma estrutura de dados do tipo pilha "p";
- POP (p): Desempilha o elemento no topo da pilha.

Considere que, em uma estrutura de dados tipo pilha "p", inicialmente vazia, sejam executados os seguintes comandos:

```
PUSH (p, 10)
PUSH (p, 5)
PUSH (p, 3)
PUSH (p, 40)
POP (p)
PUSH (p, 11)
PUSH (p, 4)
PUSH (p, 7)
POP (p)
POP (p)
```

Após a execução dos comandos, o elemento no topo da pilha "p" e a soma dos elementos armazenados na pilha "p" são, respectivamente,

- A** 11 e 29.
- B** 11 e 80.
- C** 4 e 80.
- D** 7 e 29.
- E** 7 e 40.

11. Uma transportadora precisa de um sistema para gerenciar o carregamento de caixas nos caminhões da empresa. Implemente um programa que atenda a necessidade da empresa, empregando uma pilha para definir a ordem do carregamento e descarregamento das caixas em um caminhão.

- O programa deve possuir um menu com as opções:
 - Definir quantidade de caixa – Criar uma pilha que suporte a quantidade de caixas informada.
 - Carregar caixa - empilhar
 - Descarregar caixa - desempilhar
 - Mostrar carga – mostrar pilha
 - Sair do programa

- Cada caixa possui um código e peso;

```
struct Caixa
{
    int codigo;
    float peso;
};
```

- Na estrutura pilha, a variável “dados” deve ser alterada para o tipo Caixa;

```
struct Pilha
{
    int tam;
    int base;
    int topo;
    Caixa *dados;

    Pilha()
    {
        tam=0;
        base = -1;
        topo = -1;
        dados = NULL;
    }
};
```

- Alterar as demais funções de Pilha.hpp para o tipo Caixa. Exemplos:

```
void inicializaP(Pilha *p, int tam)
{
    p->base = -1;
    p->topo = -1;
    p->tam = tam;
    p->dados = new Caixa[tam];
}

bool empilhaP(Pilha *p, Caixa c)
{
    /// retorna false se a pilha não foi inicializada ou se cheia
    if (!verificaInicializacaoP(p) || cheiaP(p))
        return false;
    else{
        p->topo++;
        p->dados[p->topo] = c;
        return true;
    }
}
```

12. A conversão de um número decimal N (base 10) para binário (base 2) é realizada por sucessivas divisões de N por 2, em que o número binário é obtido pelos restos das divisões realizadas. Crie um programa para converter número decimais em binários empregando uma pilha.
13. Crie um programa para intercalar duas pilhas P1 e P2 conforme o exemplo abaixo. Note que as pilhas P1 e P2 possuem o mesmo tamanho, mas o número de elementos armazenados em cada uma é diferente.

P1
2
3
4
5

P2
30
20
10

P3
5
10
4
20
3
30
2