

## Questão 28

### Questão 28.<sup>28</sup>

Uma estrutura de dados do tipo pilha pode ser usada em um algoritmo que permite imprimir uma palavra de forma invertida. Exemplo: FELICIDADE deve ser impresso como EDADICILEF.

Considere as variáveis declaradas abaixo.

```
pilha[1..50]: caractere;  
i, topo: inteiro;  
palavra: string;
```

Em pseudocódigo, faça o que se pede nos itens a seguir.

- A. Desenvolva a rotina push que inclui um elemento na pilha.
- B. Desenvolva a rotina pop que retira um elemento da pilha.
- C. Desenvolva a rotina que leia a palavra e, usando a pilha, a imprima de forma invertida.

## 1. Introdução teórica

### Estruturas de dados: pilhas

A estrutura de dados denominada pilha tem comportamento muito similar ao comportamento de uma pilha de objetos na vida real. Por exemplo, suponha uma pilha de livros sobre uma mesa: idealmente, podemos adicionar livros ao topo da pilha ou retirar um livro do topo da pilha. Os demais livros que se encontram abaixo do livro que está no topo não estão (diretamente) acessíveis. Essa disciplina nos processos de inserção e de remoção garante que a informação sobre a ordem de inserção seja mantida. Dessa forma, o primeiro objeto a ser inserido é o último a ser removido e o último objeto a ser inserido é o primeiro a ser removido, o que costuma ser chamado, em inglês, de *LIFO (Last In, First Out)*.

---

<sup>28</sup>Questão Discursiva 4 – Enade 2014.

## 2. Padrão de resposta INEP

### Parte A.

*A primeira parte da questão corresponde à inserção dos elementos no topo da pilha. Do enunciado, é possível perceber a sugestão do uso de um vetor estático com 50 elementos, ou seja, a pilha vai armazenar no máximo 50 elementos. Uma variável deve conter o índice do elemento que é o topo da pilha, ou seja, o último elemento a ser inserido. Essa variável deve ser incrementada cada vez que inserimos um novo elemento na pilha. É importante controlarmos o tamanho máximo, uma vez que o vetor tem apenas 50 posições. Caso o valor da variável "topo" torne-se maior do que 50, o programa deve imprimir uma mensagem de erro. A variável c contém o caractere a ser depositado no topo da pilha. Isso é ilustrado no pseudocódigo abaixo.*

```
função push(c:caractere)
início
    se topo >= 50
        escreva "Erro: Pilha Cheia"
    senão
        topo <- topo + 1
        pilha[topo] <- c
    fimse
fimfunção
```

### Parte B.

*A segunda função essencial para a implementação de uma pilha é a função pop, que retira o último elemento acrescentado ao topo da pilha. Essa função deve conter uma proteção para o caso de a pilha encontrar-se vazia, pois, nesse caso, não é possível remover nenhum elemento. Finalmente, é importante que a variável "topo" seja decrementada de uma unidade para cada remoção, uma vez que ela deve apontar para o elemento que se encontra imediatamente abaixo do elemento retirado, o qual vai ser o novo topo da pilha. O pseudocódigo a seguir ilustra essa função.*

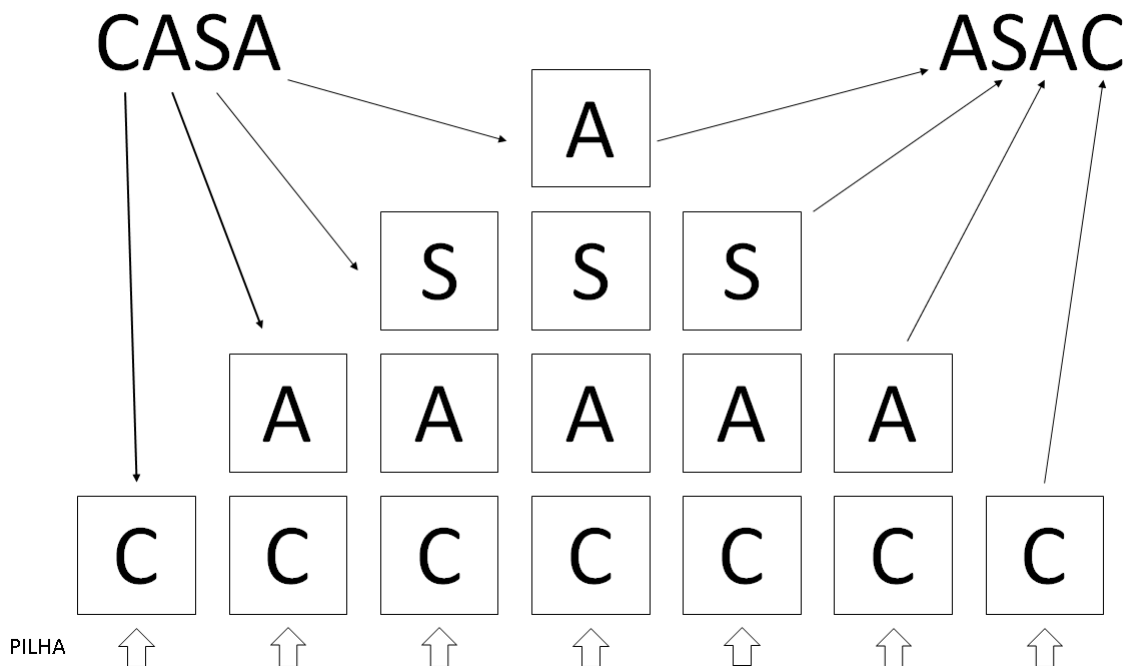
```

função pop(): caractere
início
    se topo <= 0
        escreva "Erro: Pilha Vazia"
    senão
        devolva(pilha[topo])
        topo <- topo -1
    fimse
fimfunção

```

### Parte C.

Devido ao próprio funcionamento de uma pilha, em que o último elemento a ser acrescentado é o primeiro a ser retirado, basta acrescentarmos os caracteres na pilha um a um utilizando a função push e, depois, retirá-los utilizando a função pop. Como a pilha é uma estrutura de dados do tipo LIFO, depois de depositarmos todos os caracteres de uma palavra em uma pilha, eles vão ser retirados na ordem reversa a que foram colocados, como ilustrado na figura 1.



**Figura 1.** Uma pilha com a palavra casa.

O pseudocódigo do algoritmo é ilustrado logo a seguir.

```
função inverte()  
início  
    topo <- 0  
    leia palavra  
    para i <- 1 até tamanho(palavra) passo 1 faça  
        push(palavra[i])  
    fimpara  
    para i <- 1 até tamanho(palavra) passo 1 faça  
        imprima (pop())  
    fimpara  
fimfunção
```

*Observe que o programa inicialmente lê toda a palavra, posteriormente insere essa palavra na pilha caractere a caractere (utilizando o comando push) e finalmente retira os caracteres na ordem inversa utilizando o comando pop.*

Disponível em  
<[http://download.inep.gov.br/educacao\\_superior/enade/padrao\\_resposta/2014/padrao\\_resposta\\_tecnologia\\_analise\\_desenv\\_sistemas.pdf](http://download.inep.gov.br/educacao_superior/enade/padrao_resposta/2014/padrao_resposta_tecnologia_analise_desenv_sistemas.pdf)>. Acesso em 18 set. 2017.

### 3. Indicações bibliográficas

- CELES, W.; CERQUEIRA, R.; RANGEL, J. R. *Introdução à estrutura de dados*. Rio de Janeiro: Campus, 2004.
- FEOFILLOF, P. *Algoritmos em linguagem C*. Rio de Janeiro: Elsevier Brasil, 2009.