



Início (/) > Superior de Tecnologia em Análise e Desenv... > Algoritmos e Programação Estruturada (/alu... > Aap4 - Algoritmos e Programação Estrutura...

Aap4 - Algoritmos e Programação Estruturada

Informações Adicionais

Período: 22/08/2022 00:00 à 03/12/2022 23:59

Situação: Cadastrado

Tentativas: 2 / 3

Protocolo: 816357531

A atividade está fora do período do cadastro

Avaliar Material

1) Na área da computação é comum que criemos analogias entre os conceitos e situações cotidianas, afim de facilitar o entendimento dos mesmos.

Com base nisso associe a COLUNA-A, que apresenta alguns elementos de programação, com a COLUNA-B, que apresenta alguns exemplos aplicáveis:

COLUNA-A

COLUNA-B

I. Vetor

A. Uma música em uma *playlist* criada pelo usuário.

II. Índice do Vetor

B. Edifício com 10 apartamentos.

III. Lista ligada

C. Brincadeira do amigo – secreto.

IV - Elemento de uma lista D. O número de uma casa em uma rua.

A seguir assinale a alternativa que apresenta a associação correta.

Alternativas:

a) I - B; II - D; III - C; IV - A.

☒ Alternativa assinalada

b) I - B; II - C; III - D; IV - A.

c) I - B; II - A; III - C; IV - D.

d) I - C; II - D; III - B; IV - A.

e) I - C; II - A; III - B; IV - D.

2) As listas ligadas são estruturas de dados conhecidas por serem flexíveis quanto às suas operações e seu tamanho é dinâmico. Toda lista é composta por nós, e o nó é um conjunto de informações que permitem apontar para o próximo elemento.

Considere a organização das pessoas na fila do banco, porém seu funcionamento coerente depende do comportamento da lista ligada, então aplique o conceito de inserção em listas ligadas para assinalar a alternativa correta:

Alternativas:

a) As listas só permitem inserção de elementos em seu início.

b) As listas só permitem inserção de elementos em seu fim.

c) As listas só permitem inserção de elementos em seu início ou seu fim.

☒ Alternativa assinalada

d) As listas permitem inserção de elementos em qualquer posição.

☒

e) As listas não permitem sobreposição ao inserir elementos.

3) A declaração da estrutura inicial para criação de uma pilha pode ser implementada por:

```
struct Pilha {  
  
int topo;
```

```
int capacidade;  
  
float * proxElem;  
  
};  
  
struct Pilha minhaPilha;
```

Segundo Celes, Cerqueira e Rangel (2004), com a estrutura declarada é possível criar a função para criar uma pilha. Esta função aloca dinamicamente na memória o espaço para utilização da pilha. O trecho de código para criação da pilha pode ser implementado por:

```
void cria_pilha(struct Pilha *p, int c){  
    p -> proxElem = (float*) malloc (c * sizeof(float));  
  
    p -> topo = -1;  
  
    p -> capacidade = c;  
  
}
```

Com a função para criar a pilha realizada, ela estará vazia, ou seja, não terá nenhum elemento na pilha em sua criação. Assim, é possível criar a função que vai permitir ser inserido um novo elemento na pilha.

Considerando os códigos apresentados, o **correto** formato para inserir um novo elemento na pilha usando o nome *push()* é

Alternativas:

- a)

```
void push-pilha(struct Pilha *p, float **v){  
    p -> topo++;  
    p -> proxElem [p -> topo] = v;  
}
```
- b)

```
void push_pilha(struct Pilha *p, float v){  
    p -> topo-+;  
    p -> proxElem [p -> topo] = v;  
}
```
- c)

```
void push_pilha(struct Pilha *p, float v){
```

☒
- ```
 p -> topo++;
```

```
p -> proxElem [p -> topo] = v;
}

void push_pilha(struct Pilha /p, float v){
 p -> topo++;
d) p -> proxElem [p -> topo] = v;
}

void push_pilha(struct Pilha /p, float v){
 p -> topo++;
e) p -> proxElem [p -> topo] = v;
}
```

Alternativa assinalada

4) Em uma fila, só é possível remover um elemento pelo seu início. É possível implementar o trecho de código a seguir para remoção do elemento e apresentar seu valor no retorno da função:

```
float remove_fila (Fila* f){
 char elem;
 if (fila_vazia(f)){
 printf("A Fila esta vazia\n");
 exit(1);
 }
 elem = f -> vet[f -> ini];
 f -> ini = (f -> ini + 1) % N;
 f -> n--;
 return elem;
}
```

De acordo com o trecho, marque (V) verdadeiro ou (F) falso nas asserções a seguir:

- ( ) Nesse trecho é possível observar que, antes de remover o elemento da fila, é preciso verificar se ela possui elementos, chamando a função fila vazia.
- ( ) Caso a fila esteja vazia, a função apresenta uma mensagem informando ao usuário que não há elementos para serem removidos e finaliza a função.
- ( ) Caso a fila possua elementos, a variável vet recebe o elemento da primeira posição da fila.
- ( ) Assim, o início da fila volta para o elemento anterior.
- ( ) Este elemento incrementa a quantidade de elementos e retorna o elemento removido.

Assinale a alternativa que contém a sequência correta:

**Alternativas:**

- a) V-F-V-F-V.
- b) F-V-F-V-F. Alternativa assinalada
- c) F-V-V-F-V.
- d) V-V-F-F-F. ☒
- e) V-F-F-V-F.