

Βομα

stosib oñigolo2 (4

bollen

}

e = x + 1;

**} while**

**if (v A[x]) return x;**

**else return N + 1;**

Para que isso ocorra, o trecho pontilhado no corpo da função deve ser substituído por:

a) (e > d);

b)! (v == A[x]);

(v [[A[x] || ex d);

**d) (v=A[x]**

(v != A[x] && e <= d);

6) [Questão 33 do ENADE 2017 de ADS] A ordenação por inserção funciona de modo semelhante à forma como algumas pessoas ordenam cartas de baralho. Inicia-se com a mão esquerda vazia e as cartas empilhadas na mesa. Remove-se da pilha uma carta de cada vez, inserindo-a na posição correta na mão esquerda. Para se identificar a posição correta de uma carta, deve-se compará-la com as cartas presentes na mão esquerda, no sentido da direita para a esquerda. Em todos os momentos as cartas da mão esquerda estão ordenadas, tendo sido obtidas no topo da pilha da mesa.

CORMEN, T. H. et al. Introduction to algorithms. 3. Ed Cambridge: The MIT Press, 2009 (adaptado). Um programador implementou um algoritmo de ordenação semelhante à forma de ordenação de cartas descrita no texto. Ao realizar um teste com um vetor de nove posições (vetor [1..9]), verificou que o algoritmo não funcionava corretamente.

**01 para i <- 2 até 9 faça**

valor <- vetor[i]

<- i - 1

**enquanto ((j >= 1f**

e (valor < vetor[3])) faça

vetor[i] <- vetor[i]

j<-j- 1

**se (j = 0) então**

02

03

04

05

06

07

08

09

10

**11**

vetor[j+1] <- yalor

**12 fim para**

3

**interrompa**

**fim se**

**fim enquanto**

Modulo

Com base nessas informações, assinale a opção em que se apresentam a linha e o respectivo comando a ser substituído, para que o algoritmo ordene corretamente um vetor de inteiros de forma crescente.

a) Linha 01; para i <- 1 até 9 faça

Linha 03; j <- i

Linha 04; enquanto (>= 1) ou (valor < vetor[j])) faça

Linha 05; vetor) + 1] <- vetor[j]

e) Linha 11; vetor[j] <- valor

**7) [Questão 18 do ENADE 2017 de Ciência da Computação] O algoritmo a seguir recebe um vetor v de números inteiros e rearranja esse vetor de tal forma que seus elementos, ao final, estejam ordenados de forma crescente.**

01 void ordena (int \*v, int n)

int i, j, chave;

02 {

03

04

**05**

{

**for (i = 1; i < n; i++)**