PRÁTICA 5

Questão 1) Analise o código fonte abaixo:

```
1  #include <stdio.h>
2  int main ()
3  {
4      int *p;
5      int num;
6      p=&num;
7
8      for (num=0;num<10;num++){
9          printf("\nVetor %d\n", num);
10     }
11     printf("Valor do ponteiro p=%d",*p);
12     return 0;
13 }
```

Comente as linhas 4 a 6. Mostre o que irá ser exibido na tela nas linhas 9 e 11

Questão 2) O coordenador geral de um comitê olímpico solicitou a implementação de um aplicativo que permita o registro dos recordes dos atletas à medida que forem sendo quebrados, mantendo a ordem cronológica dos acontecimentos, e possibilitando a leitura dos dados a partir dos mais recentes.

Considerando os requisitos do aplicativo, a estrutura de dados mais adequada para a solução a ser implementada é:

A) **O deque:** tipo especial de lista encadeada, que permite a inserção e a remoção em qualquer das duas extremidades da fila e que deve possuir um nó com a informação (recorde) e dois apontadores, respectivamente, para os nós próximo e anterior.
B) **A fila:** tipo especial de lista encadeada, tal que o primeiro objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura FIFO (*First In – First Out*), a inserção e a remoção são feitas em extremidades contrárias e a estrutura deve possuir um nó com a informação (recorde) e um apontador, respectivamente, para o próximo nó.
C) **A pilha:** tipo especial de lista encadeada, na qual o último objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura LIFO (*Last In – First Out*), a inserção e a remoção são feitas na mesma extremidade e a estrutura deve possuir um nó com a informação (recorde) e um apontador para o próximo nó.

D) **A fila invertida:** tipo especial de lista encadeada, tal que o primeiro objeto a ser inserido na fila é o primeiro a ser lido; nesse mecanismo, conhecido como estrutura FIFO (*First In – First Out*), a inserção é a remoção são feitas em extremidades contrárias e a estrutura deve possuir um nó com a informação (recorde) e um apontador, respectivamente, para o nó anterior.

E) **A lista circular:** tipo de lista encadeada, na qual o último elemento tem como próximo o primeiro elemento da lista, formando um ciclo, não havendo diferença entre primeiro e último, e a estrutura deve possuir um nó com a informação (recorde) e um apontador, respectivamente, para o próximo nó.

Questão 3) Faça as operações de pilha e seus efeitos sobre uma pilha S de inteiros, inicialmente vazia:

Operação	Saída	Conteúdo da pilha
push(5)	-	(5)
push(10)	-	(10,5)
push(15)		
pop( )		
pop( )		
isEmpty( )		
push(3)		
push(5)		
size( )		
pop( )		
pop( )		
size( )		

Questão 4) *A ordenação por inserção funciona de modo semelhante à forma como algumas pessoas ordem 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, 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
02    valor <- vetor[i]
03    j <- i - 1
04    enquanto ((j >= 1) e (valor < vetor[j])) faça
05       vetor[i] <- vetor[j]
06          j <- j - 1
07          se (j = 0) então
08             interrompa
09          fim se
10    fim enquanto
11    vetor[j + 1] <- valor
12 fim para
```

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`

B) Linha 3 : `j -> i`

C) Linha 4: `enquanto ((j >= 1) ou (valor < vetor [j])) faça`

D) Linha 5: `vetor [j + 1] <- vetor[j]`

E) Linha 11: `vetor[j] <- valor`