

Nome:

**Linguagem de Programação - Reavaliação**

RA:

1) Com base no programa a seguir, assinale com V as afirmações verdadeiras e com F as falsas.

**#include <stdio.h>**

int fatorial (int n)

(int cont, result;

if (n == 0)

result = 1;

*(Valor: 2,0)*

else

(

result = n;

}

}

for (cont = n 1; cont > 0; cont--)

result

result cont;

printf("O fatorial de d é %d\n", n, result);

return result;

**int main()**

1 int numero, res;

}

printf("Informe o número: \n"); scanf("%d", &numero); if (numero >= 0)

res fatorial (numero):

**return 0;**

() O programa é capaz de calcular e exibir corretamente o fatorial apenas de números naturais cujo fatorial caiba em um dado do tipo int.

() Se transferíssemos a instrução printf que está na função fatorial para a rotina main () aumentaríamos a coesão da subrotina.

() Da forma como está codificado, o programa não apresenta problemas de coesão e de acoplamento.

( ) Como o parâmetro da subrotina é tratado por valor, temos um problema de acoplamento menor do que teríamos caso o tratássemos por referência.

( ) Caso a variável numero fosse declarada fora da rotina main (ou seja, com escopo global) e a rotina fatorial também utilizasse essa mesma variável, diminuiríamos o acoplamento entre as rotinas.

2) Considere a definição de variáveis a seguir e assinale com V as afirmações verdadeiras e com Fas falsas.

(...)

**010 struct ponto (float x; float y; };**

**011 struct ponto v1;**

**012 struct ponto \*pl;**

013 float distancia;

**(...)**

( ) Apenas a variável distancia é de um tipo construído da linguagem C.

Existem 3 variáveis declaradas no trecho de código apresentado: vl, pl, distancia. Considerando os tipos de dados da struct ponto, podemos concluir que a variável v1 deve possuir exatamente 10 bytes nas implementações padrão de 32 bits da linguagem C.

() A instrução sizeof(v1) podería responder adequadamente a questão indicada no item

anterior.

() A variável v1 pode conter as coordenadas de um ponto no plano bidimensional.