NOTA 4,0

UFN Universidade Franciscana

PROVA 02 – PROGRAMAÇÃO DE SISTEMAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO
UNIVERSIDADE FRANCISCANA – UFN. 2025-02. Peso:5,0.

PROFESSOR: André F. dos Santos.
Nome do aluno: Pedro Henrique Camobarro.
Data: 02/10/25.
Nas questões de múltipla escolha, marque apenas a correta. Desligue o celular durante a prova.

1) Em relação a variáveis locais e globais, assinale a correta.
a) Variáveis locais sobrevivem após o retorno da função
b) Globais reduzem acoplamento entre módulos
c) Locais reduzem efeitos colaterais e facilitam depuração
d) Locais e globais têm o mesmo ciclo de vida ×
e) Globais são desalocadas ao terminar cada chamada ×

2) Em Assembly, o par CALL/RET atua principalmente para:
a) Mover valores entre registradores e memória
b) Empilhar parâmetros e executar aritmética
c) Salvar endereço de retorno e desviar/voltar de sub-rotinas
d) Carregar constantes no contador de programa
e) Eliminar necessidade de pilha em chamadas

3) O conteúdo típico de um frame de ativação de uma pilha de execução é:
a) Apenas variáveis globais do programa
b) Parâmetros, variáveis locais, endereço de retorno (e registradores salvos)
c) Apenas o endereço de retorno
d) Apenas os parâmetros
e) Código executável da função

4) Descreva passo a passo o que acontece na pilha quando main() chama soma(2,3) e recebe o resultado.

X 0,0 ?

?

5) Compare passagem por valor, por referência e em registradores. Cite prós/contras e um caso de uso para cada.
Por valor: definido no início do programa ex: int = 5, float
Por referência: ponteiros em C e listas em python
Por registradores: definido diretamente nos registradores da CPU

6) Sobre recursão profunda em Python (ex.: fatorial(1000) sem ajustes):
a) Conclui normalmente: pilha expande ilimitada
b) Entra em loop silencioso
c) Lança RecursionError ao exceder o limite
d) Encerra o processo sem mensagem
e) Retorna 1 por segurança

→ pro: valor definido contra: não é mutável a alterações
→ pro: deixa o código + organizado
contra: + fácil de se perder nas chamadas das variáveis

7) Qual afirmação descreve corretamente a condição de stack overflow?

a) Igual a heap overflow
b) Só ocorre em linguagens compiladas
c) Ocorre por recursão/encadeamentos que estouram o limite da pilha
d) Ocorre quando dois módulos têm variável global com o mesmo nome
e) Não está relacionada a chamadas de função

8) Se o endereço de retorno salvo pelo 'CALL' for corrompido, o mais provável é:

a) CPU corrige automaticamente
b) 'RET' é ignorado e execução segue linear
c) Salto para endereço inválido → crash/comportamento indefinido
d) Compilador impede a execução
e) Nada acontece, pois endereço de retorno é redundante

9) Observe o trecho em Python:

```
x = 2
def funcao():
  x = 10
  print(x)
funcao()
print(x)
```

Qual será a saída e o motivo?

a) 10 e 10, porque variáveis locais sempre substituem as globais
b) 2 e 2, porque a função não consegue alterar a variável local
c) 10 e 2, porque a função cria uma variável local que não altera a global
d) 2 e 10, porque a variável global prevalece em qualquer escopo
e) Erro de execução, pois o Python não diferencia escopos

10) Considere o código em Python abaixo:

```
def altera_lista(l):
  l.append(200)

def altera_num(n):
  n = n + 1

a = [1,2,3]
b = 15
altera_lista(a)
altera_num(b)
print(a, b)
```

Qual será a saída?

a) [1,2,3,200] e 11
b) [1,2,3] e 15
c) [1,2,3,200] e 15
d) [1,2,3] e 16
e) [1,2,3,200,15]