

## Mutirão de C - Prova'

Rafael, Igor, Hashi

Março 2019

### Questão 1 (0,5)

Qual é a saída do código abaixo?

```
int i = 15;
if (i > 10) {
    printf("X ");
}
else {
    printf("Y ");
}
if (i > 5) {
    printf("Z ");
}
```

- a) X Y
- b) X Z
- c) Y Z
- d) X Y Z

### Questão 2 (0,5)

Qual é a saída do código abaixo?

```
int i = 1;
int j = 1;
while (i <= 10) {
    j = i;
    i++;
}
printf("%d %d ", i, j);
```

- a) 10 9
- b) 10 10
- c) 11 10
- d) 11 11

### Questão 3 (0,5)

Qual é a saída do código abaixo?

```
int s = 1;
for (int i = 0; i <= 4; i++) {
    s += i;
}
printf("%d ", s);
```

- a) 6
- b) 7
- c) 11
- d) 16

### Questão 4 (0,5)

Considerando o código abaixo, escolha todas as alternativas corretas.

```
long soma_e_compara(long arr[], int n) {
    if (n > 0) {
        long soma = 0;
        for (int i = 0; i < n; i++) {
            soma += arr[i];
        }
        return arr[0] < soma;
    }
    return 0;
}
```

- a) ☐ soma\_e\_compara( {1, -2, 3, -4, 5}, 5 ) == 0
- b) ☐ soma\_e\_compara( {-1}, 1 ) == 0
- c) ☐ soma\_e\_compara( {10, 2, 3}, 3 ) == 0
- d) ☐ soma\_e\_compara( {6, 2, 3, 4, 5}, 3 ) == 1

### Questão 5 (0,5)

Marque todas as expressões booleanas que são verdadeiras

- a) ☐ "1" == 1
- b) ☐ 1 + 'b' == 'c'
- c) ☐ 1 + '2' == '3'
- d) ☐ 4 - '3' == '1'
- e) ☐ '4' - '3' == 1
- f) ☐ '0' == "0"

## Questão 6 (0,5)

Marque todas que são corretas

```
long conta_negativos(double mat[100][100], int linhas, int colunas) {  
    long neg = 0;  
    for (int i = 0; i < linhas; i++) {  
        for (int j = 0; j < colunas; j++) {  
            if (mat[i][j] < 0) {  
                neg++;  
            }  
        }  
    }  
    return neg;  
}
```

- a) O código retorna  sempre que for chamado como `conta_negativos(mat, 1, 1);`.
- b) O código retorna  para a matriz  $\begin{pmatrix} 1 & 2 & -1 \\ 3 & 4 & 1 \\ -1 & 3 & -1 \end{pmatrix}$
- c) O código retorna  para a matriz  $\begin{pmatrix} 1 & 2 & -1 \\ 3 & 4 & 1 \\ 0 & 3 & -1 \end{pmatrix}$
- d) O código aceitaria como entrada a matriz `double mat[99][99]`

## Questão 7 (0,5)

Considere as duas variáveis abaixo. Suponha que estão com valores válidos, ou seja, não estão com lixo de memória. Quais das alternativas fazem sentido? (pode ser nenhuma ou mais que uma)

```
int *x;  
int y;
```

- a) `*x = y`  
b) `x = &y`  
c) `*y = x`  
d) `y = &x`

## Questão 8 (0,5)

Considere o struct abaixo.

```
typedef struct {  
    int key;  
    int value;  
} entry;
```

Considere também as duas variáveis abaixo. Suponha que estão com valores válidos, ou seja, não estão com lixo de memória. Quais das alternativas fazem sentido? (pode ser nenhuma ou mais que uma)

```
entry x, y;
```

- a) `x.key = y.key`  
b) `x.key = y`  
c) `x = y.key`  
d) `x = y`

## Questão 9 (0,5)

Considere novamente o struct abaixo.

```
typedef struct {  
    int key;  
    int value;  
} entry;
```

Considere também as duas variáveis abaixo. Suponha que estão com valores válidos, ou seja, não estão com lixo de memória. Quais das alternativas fazem sentido? (pode ser nenhuma ou mais que uma)

```
entry x, *y;
```

- a) x.value
- b) x->value
- c) y.value
- d) y->value

(espaço reservado para rascunho)

## Questão 10 (1,5)

Escreva uma função que recebe um inteiro  $i$  e devolve a *quantidade de dígitos* de  $i$ . Por exemplo:

- devolve 1 se  $i$  for 5;
- devolve 2 se  $i$  for 67;
- devolve 3 se  $i$  for 890;
- e assim em diante.

Escreva também um programa que recebe um inteiro  $n$  do usuário e, para cada inteiro  $i$  entre 1 e  $n$ , imprime sua quantidade de dígitos. Esse programa deve usar a função acima.

## Questão 11 (2,0)

A *codificação rotX* é um dos sistemas de criptografia mais antigos, usado inclusive antes da invenção do computador. Ela consiste simplesmente em substituir cada letra pela letra que está  $X$  passos à frente dela na ordem alfabética.

Por exemplo, *rot5* transforma “abc” em “fgh”, pois

- “a” é substituída por “f”, que está 5 passos à frente dela na ordem alfabética;
- “b” é substituída por “g”, que está 5 passos à frente dela na ordem alfabética;
- “c” é substituída por “h”, que está 5 passos à frente dela na ordem alfabética.

Sua tarefa será criar uma função `void rot13(char str[]);` que aplica a *codificação rot13* sobre uma string. A string pode ter qualquer tipo de caractere, mas somente as letras do alfabeto, tanto maiúsculas quanto minúsculas, devem ser substituídas.

O conhecimento de códigos ASCII específicos não é necessário para responder esta questão.

## Questão 12 (2,0)

Escreva uma função

```
int converte(char code[], int *pr, int *pg, int *pb);
```

que segue a especificação abaixo.

- Recebe uma string e três apontadores para inteiros.
- Se a string for uma *cor HTML* válida, por exemplo “#90AF1E”, escreve nos inteiros os valores entre 0 e 255 que representam as intensidades de vermelho, verde e azul, respectivamente, e devolve 1.
- Senão, não escreve nada nos inteiros e devolve 0.

Uma cor HTML válida deve começar com “#”. Por simplicidade, pode supor que as letras devem ser maiúsculas.