

Desafios de Programação

Quiz 2 do Mutirão de C

Questão 1. 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 1. 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 2. 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 = y`
- B. `x.key = x.value`
- C. `x.key = y.key`
- D. `x.key = y.value`

Questão 2. 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 = y`
- B. `x.key = x.value`
- C. `x.key = y.key`
- D. `x.key = y.value`

Questão 3. 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.key
- B. x->key
- C. y.key
- D. y->key

Questão 3. 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.key`
- B. `x->key`
- C. `y.key`
- D. `y->key`