

Nome: Lucas Lemos

Matrícula: 2110013

### 1ª questão

O array `int b[2]` será armazenado na memória, reservando 2 espaços de 4 bytes cada, já que se trata de um vetor de inteiros. São alocados o número de “casas” que foi definido que esse array ocuparia, no caso 2. Desse modo, esse array ocupa um total de 8 bytes.

```
b:
0x7ffc9a3264e4 - 00
0x7ffc9a3264e5 - 00
0x7ffc9a3264e6 - 00
0x7ffc9a3264e7 - 00
0x7ffc9a3264e8 - 01
0x7ffc9a3264e9 - 00
0x7ffc9a3264ea - 00
0x7ffc9a3264eb - 00
```

O array `short b[2][3]` será alocado considerando 12 espaços necessários para todas as “casas” definidas. Esses espaços são ocupados por 6 elementos presentes no array, cada um com 2 bytes, devido ao tipo `short`.

```
a:
0x7ffc9a3264ec - 00
0x7ffc9a3264ed - 00
0x7ffc9a3264ee - 01
0x7ffc9a3264ef - 00
0x7ffc9a3264f0 - 02
0x7ffc9a3264f1 - 00
0x7ffc9a3264f2 - 03
0x7ffc9a3264f3 - 00
0x7ffc9a3264f4 - 04
0x7ffc9a3264f5 - 00
0x7ffc9a3264f6 - 05
0x7ffc9a3264f7 - 00
```

### 2ª questão

O buraco existente entre dois campos de estrutura existe devido ao alinhamento de dados. Isso representa que os dados escalares de uma estrutura são alinhados de acordo com o que ocupa o maior espaço. Desse modo, o `int a` e o `int c` definem a quantidade de bytes que cada dado ocupa. Por esse motivo, o `char b` ocupa só 1 byte, enquanto esse buraco é chamado de padding.

3ª questão

D

S

T

Q

Q

S

S

a) C1

00

00

00

04

03

02

01

02

00

00

00

00

b) 08

07

06

05

04

03

02

01

01

00

00

00

00

00

00

00

00

00

c) 04

03

02

01

01

02

00

00

d) 08

07

06

05

04

03

02

01

01

00

00

00

00

00

00

00

00

00

02

00

00

00

00

00

00

00

00

e) C1	f) 02	g) int	char
C2	01	04	cc
C3	00	03	cc
	00	02	cc
	04	01	cc
	03		
	02	h) short	char
	01	02	cc
	C1	01	cc
	C2	00	cc
	C3	00	cc
	00	00	cc
	02		
	01		
	00		
	00		