

## Laboratório 06

### Array, Estrutura e União

1. Qual o tamanho (em bytes) de cada um dos arrays abaixo e como você acha que eles são armazenados em memória?

```
int b[2];  
short a[2][3];
```

Após responder a questão acima, utilize um programa C como abaixo para ver, usando a função *dump()*, como os valores são armazenados.

```
int main() {  
    int i,j;  
    short a[2][3];  
    int b[2];  
  
    for (i = 0; i < 2; i++) {  
        b[i] = i;  
        for (j = 0; j < 3; j++)  
            a[i][j] = 3*i+j;  
    }  
  
    printf("b: \n");  
    dump(b, sizeof(b));  
    printf("a: \n");  
    dump(a, sizeof(a));  
  
    return 0;  
}
```

2. Veja a estrutura abaixo:

```
struct X {  
    int a;  
    short b;  
    int c;  
};
```

(a) Calcule o tamanho da estrutura na mão e depois faça um programa em C que mostre o tamanho dessa estrutura.

(b) Inicialize uma variável do tipo estrutura “X”, colocando valores facilmente “reconhecíveis” nos campos, para ver o armazenamento na memória, como abaixo.

```
struct X {
    int a;
    short b;
    int c;
};

struct X varx = {0xa1a2a3a4, 0xb1b2, 0xc1c2c3c4};
```

Use a função *dump()* com a variável “varx” (não com os campos, use a variável mesmo). Explique o resultado exibido.

3. Qual o tamanho e como os tipos abaixo são organizados na memória?

Obs: Depois de responder, use a função *dump()* para conferir as respostas. Uma sugestão, faça um programa de teste e atribua valores “reconhecíveis” nos campos das estruturas, como no exercício anterior.

(a)

```
struct X1 {
    char c1;
    int i;
    char c2;
};
```

(e)

```
struct X5 {
    char c1;
    char c2;
    char c3;
};
```

(b)

```
struct X2 {
    long l;
    char c;
};
```

(f)

```
struct X6 {
    short s1;
    int i;
    char c[3];
    short s2;
};
```

(c)

```
struct X3 {
    int i;
    char c1;
    char c2;
};
```

(g)

```
union U1 {
    int i;
    char c[5];
};
```

(d)

```
struct X4 {
    struct X2 x;
    char c;
};
```

(h)

```
union U2 {
    short s;
    char c[5];
};
```