

**Entrega em 18ago.** A lista pode ser respondida a lápis, desde que escrita com grafite macio. Preferencialmente, **esta** lista pode ser editada com formatação simples e em letras de 11pt, impressa e entregue em papel, ou enviada por e-mail em formato PDF.

**A lista deve ser respondida individualmente.**

Veja <http://www.inf.ufpr.br/roberto/ci210/assembly/labAssembly.pdf> para instruções sobre como instalar e usar o simulador MARS.

Para facilitar a correção indique os registradores que não são usados na convenção de chamada de funções como *re, rn, etc.*

**Ex. 1** Escreva, em binário, todos os naturais entre zero e 31, todos com cinco dígitos. Escreva a lista em quatro colunas de 8 linhas.

**Ex. 2** Para cada uma das estruturas de dados mostradas abaixo, compute seu **sizeof**, e compute o endereço de byte do elemento indicado no comentário.

```
int a; // sizeof( a ) ?
int x[NX], y[NY]; // sizeof( x ) ?
short z[NZ]; // sizeof( z ) ?
char w[NW]; // sizeof( w ) ?
...
a=x[ y[ 3 ] ]; // sizeof( y[3] ) ? endereço de x[y[3]] ?
...
typedef elemType { // sizeof( elem ) ?
    elemType *next; // sizeof( next ) ?
    int vet[5]; // sizeof( vet ) ?
} elem;
elemType strut[256]; // endereço de strut[i].vet[j] ?

typedef struct A {
    int x; // sizeof( x ) ?
    short s[2] // sizeof( s ) ?
    char c[4]; // sizeof( c ) ?
} aType; // sizeof( aType ) ?
aType V[1024]; // endereço de V[i].c[2] ?
// sizeof( V ) ?
```

**Ex. 3** Traduza para *assembly* do MIPS o trecho de programa abaixo. Seu código *assembly* deve empregar as convenções de programação do MIPS. Em C, o operador de ou-exclusivo é o caractere circunflexo (^).

```
int fun(int a, int b, int c, int d, int e, int f, int g);
...
x = fun(16*a, (int)z*w, i=p^q, z=v[3], v[3*i], z=z-2, v[z+2]);
...
```

continua...

**Ex. 4** Traduza para *assembly* do MIPS os três trechos de programa em C listados abaixo. Seu código deve empregar as convenções do *assembly* do MIPS.

```
// (a)
#define N 256
#define M 64
int a,b,i;
int x[N], y[M];
...
a = x[i] + x[ y[3] % N ]; // não use divisão para calcular % (módulo)
i = y[i+1] / 4;
b = x[i % N] + x[ ( y[ (2*i)%M ] ) % N ];
```

```
// (b)
int log2(int n) {
    if (n < 2) then
        return 0;
    else
        return (1 + log2(n/2));
}
...
x = log2(96000);
...
```