

1) Mostre o código MIPS necessário para implementar os seguintes comandos em C:

```
int a,b,i;
int x[NNN], y[MMM];
a = x[10] + x[ y[3] ];
i = a/4;
b = x[i] + x[ y[2*i] ];
```

2) Traduza para *assembly* do MIPS o laço abaixo:

```
typedef struct A {                                     // compilador aloca V em 0x800000
    int x;                                              aType V[1024];
    int y;                                              int i,j,a,b,c;
    short z;
    short w;
    char s[4];
} aType;

a=b=c=0;
for (i=0; i<1024; i++) {
    a = a + V[i].x + (int)V[i].z;
    b = b + (int)(V[i].s[0] + V[i].s[3]);
    c = c + V[i].y - (int)V[i].w;
}
```

3) Traduza para *assembly* do MIPS os laços abaixo:

```
L1: for (a=0,i=0; i<10; i++) {
    a = a + i;
}

L2: for (a=0,i=0; i<k; i++) {
    a = a + i;
}

L3: a=i=0;
while (i < k) {
    a = a + i;
    i = i + 1;
}
```

4) Traduza para *assembly* do MIPS as quatro funções abaixo:

```
int fat1(int n) { // iterativo
    int i,j;
    j=1;
    if(n > 1)
        for(i=1; i <= n; i++)
            j = j*i;
    return(j);
}

int fat2(int n) { // recursivo
    if (n > 0)
        return (n * fact(n - 1));
    else
        return (1);
}

int fun(int g, int h, int i, int j, int k) {
    int f;
    f = (g+h)-(i+j)*k;
    return (f*4);
}

clear1(int array[], int size) { // vetor
    int i;
    for (i=0; i < size; i++)
        array[i]=0;
}

clear2(int *array, int size) { // apontador
    int *p;
    for (p=array; p < &(array[size]); p++)
        *p = 0;
}
```