

**Ex. 1** Traduza para *assembly* do MIPS os laços abaixo.

```
// (a) -----  
L1: for (a=0,i=0; i<10; i++) {  
    a = a + i;  
}  
  
// (b) -----  
L2: for (a=0,i=0; i<k; i++) {  
    a = a + i;  
}  
  
// (c) -----  
L3: a=i=0;  
while (i < k) {  
    a = a + i;  
    i = i + 1;  
}  
  
// (d) -----  
  
int Strcopy(const char *y, char *x) // copia y sobre x  
{  
    int i=0;  
    while ( (*x++ = *y++) != '\0' ) // copia e testa final  
        i = i+1;  
    return(i+1);  
}  
  
// (e) -----  
  
int Strcat(char *si, const char *sf) // concatena si após sf  
{  
    char *p = si;  
    int n=0;  
    while (*p != '\0') {  
        ++p;  
        n++;  
    }  
    while ( (*p++ = *sf++) )  
        n++;  
    return(n+1);  
}  
  
// (f) -----  
  
int Strrev(char *f, char *d, int n) { // reverte N caracteres de f  
    int i;  
    d = (char *) ( (int)d+n );  
    for (i=0; (i < n) && (*f != '\0') ; i++)  
        *d-- = *f++;  
    return i;  
}
```

**Ex. 2** Traduza para *assembly* do MIPS as funções abaixo.

```
// (a) -----  
int a, i;  
int x[2048], y[64];  
...  
i=1;  
a=0;  
while (i < 2048) {  
    a = a + x[i] + x[ y[i%64] % 2048 ]; // MOD, MOD  
    i = i * 2;  
}
```

```
// (b) -----  
...  
x = power(y,z);  
...  
int power(int n,int exp) {  
    if (exp > 1)  
        return (n * power(n,exp-1));  
    else  
        return (n);  
}
```

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

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

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