Tradução C / Assembly – Exemplo 1

```c
#define SIZE 25
double average(double *, int);

void main(void)
{
   double array[SIZE];
   double avg;
   ...
   avg = average( array, SIZE );
   print_double( avg );     // syscall 3
}
```

```c
double average(double *v, int N)
{
   double sum = 0.0;
   int i;

   for(i = 0; i < N; i++)
      sum += v[i];
   return sum / (double)N;
}
```

Conversão entre tipos (inteiro para double)

Tradução C / Assembly – Exemplo 1

```c
void main(void)
{
   static double array[SIZE];
   double avg;
   ...
   avg = average( array, SIZE );
   print_double( avg );     // syscall 3
}
```

```
double average(double *, int)
```

```
        .data
array:  .space 200              # 8*SIZE (alinhado múltiplo 8)
        .eqv SIZE,25
        .text
        .globl main             # avg: $f12
main:   ...                     # Salvaguarda $ra
        la      $a0, array      #
        li      $a1, SIZE       #
        jal     average         #
        mov.d   $f12, $f0       # avg = average(array, SIZE)
        li      $v0, 3          #
        syscall                 # print_double(avg)
        ...                     # Repõe $ra
        jr      $ra             #
```

Tradução C / Assembly – Exemplo 1

```c
double average(double *v, int N)
{
   double sum = 0.0;
   int i;
   for(i = 0; i < N; i++)
      sum += v[i];
   return sum / (double)N;
}
```

```
# sum: $f0  /  tmp1: $f4  /  i: $t0  /  tmp2: $t1
average: mtc1     $0,  $f0          #
         cvt.d.w  $f0, $f0          # sum = 0.0
         li       $t0, 0            # i = 0
for:     bge      $t0, $a1, endf    # while(i < N) {
         sll      $t1, $t0, 3       #   tmp = i * 8
         addu     $t1, $t1, $a0     #   $t1 = &v[i]
         l.d      $f4, 0($t1)       #      $f4 = v[i]
         add.d    $f0, $f0, $f4     #   sum += v[i]
         addi     $t0, $t0, 1       #      i++
         j        for               # }
endf:    mtc1     $a1, $f4          #
         cvt.d.w  $f4, $f4          # $f4 = (double)N
         div.d    $f0, $f0, $f4     # return sum / (double)N
         jr       $ra               #
```