

## Código:

Programa em c	Assembly
<pre>int mais_as(char *s){  int i, contaA, conta;     i=0;     contaA=0;     conta= 0;     while ( s[i] != '\0' ) {         if (s[i]== 'A'){             contaA= contaA + 1;             conta= conta +1;             i++;         }         else {             conta= conta +1;             i++;         }     }      if (contaA&gt;= conta/2)         return 1;     else {         return 0;     } }</pre>	<pre>0x08048560 &lt;+0&gt;: push    %esi 0x08048561 &lt;+1&gt;: mov     \$0x1,%eax 0x08048566 &lt;+6&gt;: push    %ebx 0x08048567 &lt;+7&gt;: xor     %ecx,%ecx 0x08048569 &lt;+9&gt;: mov     0xc(%esp),%esi 0x0804856d &lt;+13&gt;: jmp     0x80 4857e &lt;mais_as+30&gt; 0x0804856f &lt;+15&gt;: nop 0x08048570 &lt;+16&gt;: cmp     \$0x41,%dl 0x08048573 &lt;+19&gt;: sete    %dl 0x08048576 &lt;+22&gt;: add     \$0x1,%eax 0x08048579 &lt;+25&gt;: movzbl  %dl,%edx 0x0804857c &lt;+28&gt;: add     %edx,%ecx 0x0804857e &lt;+30&gt;: movzbl  -0x1(%esi,%eax,1),%edx 0x08048583 &lt;+35&gt;: lea     -0x1(%eax),%ebx 0x08048586 &lt;+38&gt;: test    %dl,%dl 0x08048588 &lt;+40&gt;: jne     0x8048570 &lt;mais_as+16&gt; 0x0804858a &lt;+42&gt;: sar     %ebx 0x0804858c &lt;+44&gt;: xor     %eax,%eax 0x0804858e &lt;+46&gt;: cmp     %ebx,%ecx 0x08048590 &lt;+48&gt;: setge   %al 0x08048593 &lt;+51&gt;: pop     %ebx 0x08048594 &lt;+52&gt;: pop     %esi 0x08048595 &lt;+53&gt;: ret</pre>

## Análise do código:

<u>c</u>	<u>Assembly</u>
*s	(%esp),%esi
i	%eax
'A'	\$0x41
conta	%ebx
contaA	%ecx
s[i]	%edx
while ( s[i] != '\0' )	jmp 0x80 4857e <mais_as+30> test %dl,%dl jne 0x8048570 <mais_as+16>
s[i]== 'A'	cmp %ebx,%ecx sete %dl
conta/2	sar %ebx
if (contaA>= conta/2) return 1; else { return 0; }	cmp %ebx,%ecx setge %al

## Tabela de alocação de registos:

<u>c</u>	<u>Assembly</u>
*s	(%esp),%esi
i	%eax
conta	%ebx
contaA	%ecx
s[i]	%edx