

SC TPC 9 - André Filipe Novais Vaz - A93221

```
(1) int foor (int a, int *b) {  
    int i = 0; j = 0;  
    for (; i < a; i++) {  
        if (b[i] > 1000) j += b[i];  
    }  
    return j;  
}
```

(2) a) %eae - j - trata-se da soma final,
valor ~~retornado~~ devolvido pela função
%eae - i - variável ~~temporária~~ para cache
%ebe - a - dimensão da lista
%esi - b - lista de números

② c)

Soma - grandes:

```
pushl %ebp
movl %esp, %ebp
pushl %ebx - guarda o registro de %ebx
pushl %esi - guarda o registro de %esi
movl 8(%ebp), %esi - define %esi igual ao primeiro arg
movl 12(%ebp), %ebx - define %ebx igual ao segundo arg
xorl %eax, %eax
xorl %ecx, %ecx } inicio as variáveis i e j
```

.foo:

```
cmpl %esi, %ecx -> ciclo for para percorrer a
                    lista
jge .jee
movl (%ebx, %ecx, 4), %edx -> %edx = b[i]
cmpl $1000, %edx -> detetar termino do for loop
jle .for
addl %edx, %eax -> adicionar o número em b[i]
```

.for:

```
incl %ecx -> i++
jmp .foo -> percorrer o ciclo for
```

.jee:

```
popl %esi
popl %ebx
leave
ret } recupera os registos dos números
```


(2)

Jobber:

pushl	%ebp
movl	%esp, %ebp
pushl	%esi
pushl	%ebx
xorl	%eax, %eax
movl	8(%ebp), %ebx
xorl	%ecx, %ecx
cmpl	%ebx, %eax
movl	12(%ebp), %esi
jge	.L9
pldigm	2, 3

— Differences

.L7:

movl	(%esi, %ecx, 4), %ecx
cmpl	\$1000, %ecx
jle	.L9
addl	%ecx, %eax

.L9:

incl	%ecx
cmpl	%ebx, %ecx
jle	.L7

— Differences

.L9:

popl	%ebx
popl	%esi
leave	
ret	

③

A sua função só testada 3 vezes !!

Teste 0: 20 elementos . Resultado correto = 64469;

Resultado obtido = 64469!

Teste 1: 50 elementos . Resultado correto = 255581;

Resultado obtido = 255581!

Teste 2: 100 elementos . Resultado correto = 500123;

~~Resultado~~ Resultado obtido = 500 123!

Teve sucesso 3 vezes !!

That's all, folks!