

Nº: a92839

NOME: BEATRIZ SOUSA DEMÉTRIO

TURMA: PL2

①

```

int soma - grandes (int m, int *a)
{
    int i;
    int result;
    result = 0;
    for (i=0; i<m; i++)
    {
        if (a[i] > 1000)
        {
            result += a[i]
        }
    }
    return result;
}
    
```

②

a)

UTILIZAÇÃO DOS REGISTOS		
VARIÁVEL	REGISTO	COMENTÁRIOS
m	% esi	calde varred (não argumentos)
a	% ebx	calde varred (não argumentos)
i	% ecx	registro que veio a seguir escolhido
a[i]	% edx	registro que veio a seguir escolhido
return	% eax	valor que é retornado (a soma)

b) ??

ENDEREÇO	CONTEÚDO	COMENTÁRIO
- 8 (%ebp) →	-----	salvaguarda de % esi
- 4 (%ebp) →	-----	salvaguarda de % ebx
bp: (%ebp) →	-----	frame pointer da função chamada
4 (%ebp) →	-----	endereço de retorno

c)

push	%ebp	# começa a função
mov	%esp, %ebp	# esp = ebp
push	%esi	# salva o endereço de esi
push	%ebx	# salva o endereço de ebx
mov	0xc(%ebp), %ebx	# a* → ebx
mov	0x8(%ebp), %esi	# m → esi
xor	%eax, %eax	} coloca os registros eax e ecx a zero, ou seja, i = 0 e result = 0.
xor	%ecx, %ecx	
cmp	%ecx, %esi	# compara i com m para controle do ciclo
jge	"final"	
mov	(%ebx, %ecx, 4), %edx	
cmp	\$0x3e8, %edx	# 1000 em hexadecimal 0x3e8 (comparação)
jle	"incremento do i"	
add	%edx, %eax	# soma o valor com o resultado
inc	%ecx	# incrementa ecx
jp	"para onde diz a instrução"	
pop	%ebx	
pop	%esi	
leave		
ret		

d) ??

000000 < soma - grandes >:

0:	55	push	%ebp	
1:	89 25	mov	%esp, %ebp	
3:	56	push	%esi	
4:	53	push	%ebx	
5:	31 c0	xor	%eax, %eax	
7:	8b 5d 08	mov	0x8(%ebp), %ebx	} instrução trocada a ordem
a:	31 c9	xor	%ecx, %ecx	
c:	39 d8	cmp	%ebx, %eax	
e:	8b 75 0c	mov	0xc(%ebp), %esi	
11:	7d 13	jge	26 < soma - grandes + 0x26 >	
13:	90	nop	→ não coloquei	
14:	8b 14 8e	mov	(%esi, %ecx, 4), %edx	
17:	81 ba e8 03 00 00	cmp	\$0x3e8, %edx	
1d:	7e 02	jle	21 < soma - grandes + 0x21 >	
1f:	01 d0	add	%edx, %eax	
21:	41	inc	%ecx	
22:	39 d9	cmp	%ebx, %ecx	} não coloquei instrução
24:	7c ee	jle	14 < soma - grandes + 0x14 >	
26:	5b	pop	%ebx	
27:	5e	pop	%esi	
28:	c9	leave		
29:	c3	ret		