

**Vinicius Koblinski de Moraes, Mateus Antonio Daniel**

**Grade de um valor:**

inicio:

mov eax, 6

mov edi, 4

caso1:

mov ebx, 10

cmp eax, ebx

jnz caso2

mov edi, 1

jmp fim

caso2:

mov ebx, 9

cmp eax, ebx

jnz caso3

mov edi, 2

jmp fim

caso3:

mov ebx, 8

fim:

**Determina idade em anos (EAX=ano, EBX=mes, ECX=dia, resultado em EDI):**

inicio:

mov eax, 2000

mov ebx, 1

mov ecx, 17

mov edi, 2024

sub edi, eax

```
dec edi
aniversario:
mov esi, 2
cmp ebx, esi
jz mes
cmp ebx, esi
jl jafez
jmp fim
mes:
mov esi, 26
cmp ecx, esi
jle jafez
jmp fim
jafez:
inc edi
jmp fim
fim:
```

**Eleva um número a uma potência (EAX=numero, EBX=potencia, resultado em EDI)**

```
inicio:
mov eax, 10
mov ebx, eax
mov ecx, 4
mov esi, 1
repeticao:
mul ebx
dec ecx
cmp ecx, esi
jnz repeticao
jmp fim
fim:
```

### **Número primo:**

inicio:

mov ebx, 22

mov ecx, 0

mov esi, ebx

mov edi, 1

repeticao:

mov edx, 0

mov eax, ebx

dec esi

div esi

cmp edx, ecx

jnz repeticao

cmp esi, edi

jz fim

mov edi, 0

fim:

### **Calcula a quantidade de divisores de um número (EAX=número, resultado em EDI)**

inicio:

mov ebx, 1

mov edi, 1

mov ecx, 0

repeticao:

mov eax, 11

cmp, eax, ebx

jz fim

div ebx

inc ebx

cmp, edx, ecx

jz alegria

jmp repeticao

alegria:  
add edi, 1  
jmp repeticao  
fim:

**Calcula se um determinado numero é impar (EDI =1 se IMPAR)**

inicio:  
mov eax, 10  
mov ebx, 2  
mov edi, 0  
mov ecx, 0  
div ebx  
cmp, edx, ecx  
jz alegria  
add edi, 1  
jmp fim  
alegria:  
jz fim  
fim:

**Verifica se um número é um número perfeito (EAX=número, atribui 1 a registrador EDI se sim)**

inicio:  
mov eax, 28  
mov ebx, eax  
mov ecx, 0  
mov esi, 0  
repeticao:  
mov eax, 28  
mov edx, 0  
dec ebx

```
cmp ebx, ecx
jz verificacao
div ebx
cmp edx, ecx
jnz repeticao
soma:
add esi, ebx
jmp repeticao
verificacao:
cmp esi, eax
jnz fim
mov edi, 1
fim:
```

### **Maior Divisor Comum:**

```
inicio:
mov eax, 18
mov ebx, 12
mov ecx, ebx
mov esi, eax

repeticao:
mov eax, esi
mov edx, 0
div ecx
cmp edi, edx
jnz decrementar

mov edx, 0
mov eax, ebx
div ecx
cmp edi, edx
```

jnz decrementar

mov edi, ecx

jmp fim

decrementar:

dec ecx

jmp repeticao

fim: