

# Código Equivalente em C

---

```
int primoAtual = 2;
int RAM[128];

int main()
{
    verifica_primos();
    if(primoAtual == 2)
    {
        primoAtual = 3;
        verifica_primos();
    }
    else if (primoAtual == 3)
    {
        primoAtual = 3;
        verifica_primos();
    }
    else
    {
        primoAtual = 5;
        verifica_primos();
    }

    for(int i = 0; i < 33; i++)
    {
        printf("%i", RAM[i]); //Em R3
    }
}

void verifica_primos()
{
    for (int i = 0; i < 33; i++)
    {
        int numeroAtual = RAM[i];

        while (numeroAtual > 0)
        {
            numeroAtual -= primoAtual;
        }

        //Se o número atual ficou exatamente zero,
        //significa que não é primo
        if (numeroAtual == 0)
```

```
        RAM[i] = 0;
    }
}
```

## Código Equivalente em Assembly

---

```
ld %r5, 1
ld %r1, 33

loop_carrega_memoria:
ld [r2] %r2
add %r2, 5
jplt loop_carrega_memoria

ld %r0, 2
ld %r2, 3

for_inicio:
cmp %r2, %r1
jpgt for_fim

ld %r4 [%r2]

while_inicio:
cmp %r4, %r7
brle fim_while

sub %r4, %r0
ld %r6, while_inicio
jpa %r6

while_fim:

cmp %r4, %r7
jrne primo

ld [%r2], %r7

primo:

add %r2, %r5

ld %r6, for_inicio
jpa %r6
```

```
cmp %r0, 2
jrne primo_3

ld %r0, 3
ld %r2, 4
jpa for_inicio

primo_3:

cmp %r0, 3
jrne primo_5
ld %r0, 5
ld %r2, 6

jpa for_inicio


ld %r2, 0

iteracao_resultados:
ld %r3, [%r2]

add %r2, %r5

cmp %r2, 33

jrslt iteracao_resultados

loop_infinito:
cmp %r7, 0

jeq loop_infinito
```

## Fluxograma

---

