

Estruturas de Repetição

Programação e Desenvolvimento de Software I

Até agora...

1. Sabemos escrever programas de forma linear
2. Sabemos usar comandos condicionais para tomar decisões
 - a. Qual bloco executar
 - b. if/else
 - c. switch
3. Na primeira aula calculamos a divisão de dois números usando somas e subtrações. Para isto precisamos de comandos de repetição.

while

while; enquanto

- Estrutura do comando while

```
while ( condicao ) comando;  
while ( condicao ) { comandos }
```

- Enquanto a condição for verdadeira ($\neq 0$), ele executa o(s) comando(s);

Problema

- Vamos supor que só sabemos fazer operações de:
 - Soma (+)
 - Subtração (-)
- Dado dois inteiros **A** e **B**
- Determine o **quociente** e o **resto** da divisão de A por B.

Problema

- Vamos supor um que só sabemos fazer operações de:
 - Soma (+)
 - Subtração (-)
- Dado dois inteiros **A** e **B**
- Determine o **quociente** e o **resto** da divisão de A por B
- **Já vimos isso antes?!**

Nosso velho algoritmo

```
#include <stdio.h>
int main() {
    int A = 27;
    int B = 4;
    int q = 0;
    int r = 0;
    while (A >= B) {
        A = A - B;
        q += 1;
    }
    r = A;
    return 0;
}
```

<http://goo.gl/8sKZdj>

- Enquanto $A \geq B$
 - Entre no bloco

Nosso velho algoritmo

```
#include <stdio.h>
int main() {
    int A = 27;
    int B = 4;
    int q = 0;
    int r = 0;
    while (A >= B) {
        A = A - B;
        q += 1;
    }
    r = A;
    return 0;
}
```

<http://goo.gl/8sKZdj>

- Enquanto $A \geq B$
 - Entre no bloco
- No momento em que $A < B$
 - saia do bloco
 - continue o programa

Problema

Imprimir os 100 primeiros números naturais

Problema

Imprimir os 100 primeiros números naturais

```
#include <stdio.h>
int main(void) {
    int natural = 1;
    while (natural <= 100) {
        printf("%d\n", natural);
        natural++;
    }
    return 0;
}
```

$\ast =$, $+=$, $-=$, $/=$, $++$, $--$

`var1 = var1 * var2; \leftrightarrow var1 *= var2;`

`var1 = var1 / var2; \leftrightarrow var1 /= var2;`

`var1 = var1 + var2; \leftrightarrow var1 += var2;`

`var1 = var1 - var2; \leftrightarrow var1 -= var2;`

`var1++ \leftrightarrow var1 += 1 \leftrightarrow var1 = var1 + 1;`

`var1-- \leftrightarrow var1 -= 1 \leftrightarrow var1 = var1 - 1;`

Problema

Imprimir as 10 primeiras potências de 2

Problema

Imprimir as 10 primeiras potências de 2. Passo a passo → <https://goo.gl/ABiVg4>

```
#include <stdio.h>
int main(void) {
    int potencia = 1;
    int i = 1;
    while (i <= 10) {
        printf("%d\n", potencia);
        potencia *= 2;
        i += 1;
    }
    return 0;
}
```

Problema

Máximo divisor comum (MDC) entre dois números. Por exemplo:

Divisores de 30 → 1, 2, 3, 5, 6, 10, 15, 30

Divisores de 12 → 1, 2, 3, 4, 6, 12

Resposta é 6!

```
#include <stdio.h>
int main(void) {
    int v1, v2;
    printf("Digite dois valores: ");
    scanf("%d %d", &v1, &v2);

    int i = 1;
    int r = 0;
    while (i <= v1 && i <= v2) {
        if ((v1 % i == 0) && (v2 % i == 0))
            r = i;
        i++;
    }
    printf("O mdc entre %d e %d foi %d\n", v1, v2,
r);
    return 0;
}
```

```
#include <stdio.h>
int main(void) {
    int v1, v2;
    printf("Digite dois valores: ");
    scanf("%d %d", &v1, &v2); Ler 2 números

    int i = 1;
    int r = 0;
    while (i <= v1 && i <= v2) {
        if ((v1 % i == 0) && (v2 % i == 0))
            r = i;
        i++;
    }
    printf("O mdc entre %d e %d foi %d\n", v1, v2,
r);
    return 0;
}
```

```
#include <stdio.h>
int main(void) {
    int v1, v2;
    printf("Digite dois valores: ");
    scanf("%d %d", &v1, &v2);

    int i = 1;
    int r = 0;
    while (i <= v1 && i <= v2) {
        if ((v1 % i == 0) && (v2 % i == 0))
            r = i;
        i++;
    }
    printf("O mdc entre %d e %d foi %d\n", v1, v2,
r);
    return 0;
}
```

Solução:

Incrementar i até que $v1$ e $v2$ sejam divisíveis por i

for

for

- Estrutura do comando for

```
for (início; condição; passo) comando;  
for (início; condição; passo) { comandos }
```

- **início**: Uma ou mais atribuições, separadas por “,”
- **condição**: Idêntico ao comando while
- **passo**: Um ou mais comandos, separados por “,”

Problema

Vamos fazer um programa que imprime todos os números de 0 até 99

Comando **for**

for (início; condição; passo)

```
#include <stdio.h>
int main(void) {
    for (int i = 0; i <= 99; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```

Problema

Imprimir os números pares até um número máximo dado

Números pares até um máximo

Passo a passo

```
#include <stdio.h>

int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }

    return 0;
}
```

Números pares até um máximo

Lemos com scanf

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
}
return 0;
```

	max	i	i < max	i % 2
Antes for	5			

for (início; condição; passo)
i = 0 é a inicialização

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0		

```
for (início; condição; passo)
se a condição de entrada TRUE
```

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	

Bloco do for sendo executado

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE

Bloco do for sendo executado

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
}
return 0;
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE

for (início; condição; passo)
após terminar o for vamos para o

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1		

for (início; condição; passo)

vemos a condição novamente

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	

```
for (início; condição; passo)
ainda válida, então entramos no
```

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE

for (início; condição; passo)
o if foi false, for termina (não há mais comandos).

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
    {
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE
Iteração 2		2		

for (início; condição; passo)
assim por diante

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE
Iteração 2	5	2	TRUE	TRUE
Iteração 3	5	3	TRUE	FALSE
Iteração 4	5	4	TRUE	TRUE

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número:");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
    {
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

for (início; condição; passo)
acabamos de terminar o $i = 4$.

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE
Iteração 2	5	2	TRUE	TRUE
Iteração 3	5	3	TRUE	FALSE
Iteração 4	5	4	TRUE	TRUE
	5	5		

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número:");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
    {
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

```
for (início; condição; passo)
```

i < max → FALSO

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE
Iteração 2	5	2	TRUE	TRUE
Iteração 3	5	3	TRUE	FALSE
Iteração 4	5	4	TRUE	TRUE
Iteração 5	5	5	FALSE	

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

for (início; condição; passo)
saímos do for!

	max	i	i < max	i % 2
Antes for	5			
Iteração 0	5	0	TRUE	TRUE
Iteração 1	5	1	TRUE	FALSE
Iteração 2	5	2	TRUE	TRUE
Iteração 3	5	3	TRUE	FALSE
Iteração 4	5	4	TRUE	TRUE
Iteração 5	5	5	FALSE	

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i++)
    {
        if (i % 2 == 0)
            printf("%d\n", i);
    }
    return 0;
}
```

Imprimindo números pares

- O passo não precisa necessariamente ser `i++`
- Como vai ser a saída do programa ao lado?

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);

    for (int i = 0; i < max; i+=2) {
        printf("%d\n", i);
    }
    return 0;
}
```

Imprimindo números pares

- O passo não precisa necessariamente ser `i++`
- Como vai ser a saída do programa ao lado?
- E dessa nova variação?

```
#include <stdio.h>
int main(void) {
    int max;
    printf("Digite um número: ");
    scanf("%d", &max);
    int i = 0;
    for (; i < max; i+=2) {
        printf("%d\n", i);
    }
    return 0;
}
```

Problema

Faça um programa que computa a potência de um número (v1) elevado a um outro número (v2)

Problema

Faça um programa que computa a potência de um número (v1) elevado a um outro número (v2)

```
#include <stdio.h>
int main(void) {
    int v1, v2;
    int r = 1;
    printf("Digite dois valores: ");
    scanf("%d %d", &v1, &v2);

    for (int i = 1; i <= v2; i++)
        r *= v1;

    printf("%d^%d = %d\n", v1, v2, r);
    return 0;
}
```

do while

do-while; faça até

```
do comando; while ( condicao );
```

```
do { comandos } while ( condicao );
```

// Diferente do while, sempre entramos no bloco a primeira vez

Problema

Ler dois números e computar a potência deles. Em seguida, perguntar se deseja calcular novamente.

Problema

Ler dois números e computar a potência deles. Em seguida, perguntar se deseja calcular novamente.

```
#include <stdio.h>
int main(void) {
    int v1, v2, resposta;
    int r;
    do {
        printf("Digite dois valores: ");
        scanf("%d %d", &v1, &v2);
        r = 1;
        for (int i = 1; i <= v2; i++)
            r *= v1;
        printf("%d^%d = %d\n", v1, v2, r);
        printf("Deseja repetir? Sim (1) ou nao
(0)\n");
        scanf("%d", &resposta);
    } while (resposta);
}
```

Notas Finais

- Todos os laços tem um propósito comum
- Qualquer programa feito com um laço pode ser feito com outro
 - todos são equivalentes
- Pequeno diferencial simplifica alguns casos.
 - Tente fazer o último problema com for e while

Exercício

Faça um programa que imprima todos os números **primos** de 0 até 99. Use qualquer laço que achar mais adequado.

MDC - Algoritmo de Euclides

```
#include <stdio.h>
int main(void) {
    int v1, v2, r;
    printf("Digite dois valores: ");
    scanf("%d %d", &v1, &v2);

    r = v1 % v2;
    while (r != 0) {
        v1 = v2;
        v2 = r;
        r = v1 % v2;
    }
    printf("O mdc eh %d e %d foi %d\n", v2);
    return 0;
}
```