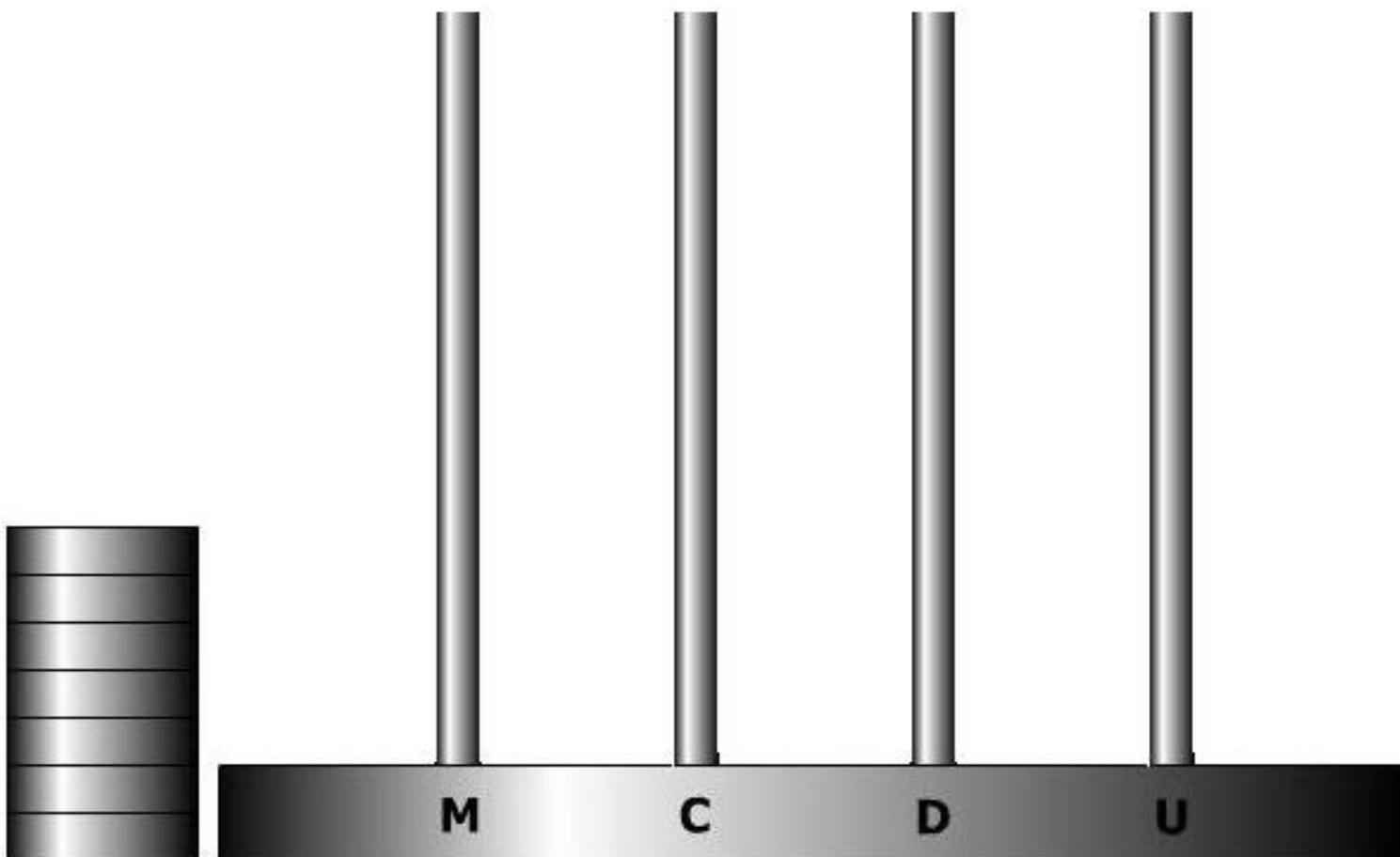


Controle de Fluxo



Através das estruturas de controle de fluxo de execução abaixo poderemos criar algoritmos para solucionar qualquer problema

- Seleção Simples
- Seleção Encadeada
- Seleção Composta
- Repetição com teste no início
- Repetição com teste ao final
- Repetição com variável de controle

Seleção Simples

- Uma estrutura de seleção simples permite a escolha de um grupo de ações e estruturas a ser executado quando determinadas condições, representadas por expressões lógicas, são ou não satisfeitas

```
int idade = leInteiro("Informe sua Idade");  
  
if(idade >= 18) {  
    escrevaL("Você é maior de idade");  
}
```

Seleção Composta

- Na seleção composta, caso o resultado da condição seja falsa, teremos a execução de uma outra seqüência de comandos.

```
int idade = leInteiro("Informe sua Idade");  
  
if(idade >= 18) {  
    escrevaL("Você é maior de idade");  
} else {  
    escrevaL("Você é menor de idade");  
}
```

Seleção Composta

- Na seleção composta também permite que seja encadeado uma seqüência de testes

```
int idade = leInteiro("Informe sua Idade");

if(idade < 14) {
    escrevaL("Você ainda é uma criança");
} else if(idade < 18) {
    escrevaL("Você é quase maior de idade");
} else {
    escrevaL("Você é maior de idade");
}
```

Seleção Encadeada

- Uma outra forma de tratar um problema de lógica com seleções encadeadas, quando a ação a ser executada depende do valor de uma variável

```
int codigo = leInteiro("Informe o código de acesso");

switch (codigo) {
case 1:
    escrevaL("Vá para o quinto andar");
    break;
case 3:
    escrevaL("Vá para o nono andar");
    break;
default:
    escrevaL("Vá para o primeiro andar");
    break;
}
```

Repetição com teste no início

- Consiste numa estrutura de controle do fluxo lógico que permite executar diversas vezes um mesmo trecho do algoritmo, porém, sempre verificando antes de cada execução se é “permitido” repetir o mesmo trecho.

```
int quantidade = leInteiro("Informe a quantidade de valores");

double total = 0;
int contador = 1;
while (contador <= quantidade) {
    double valor = leReal("Informe o ", contador, "º valor");
    total = total + valor;
    contador = contador + 1;
}
escrevaL("O valor Total é de R$", total);
```

Repetição com teste ao final

- Para realizar a repetição com teste no final, utilizamos a estrutura “repita”, que permite que um bloco ou ação primitiva seja repetida até que uma determinada condição seja verdadeira.

```
double total = 0;
int contador = 1;
char continua = 'N';
do {
    double valor = leReal("Informe o ", contador, "º valor");
    total = total + valor;
    contador = contador + 1;
    continua = leCaracter("Deseja continuar? (informe S ou N)");
} while (continua == 'S');
escrevaL("O valor Total é de R$", total);
```


Repetição com variável de controle

- A estrutura “para” repete a execução do bloco um número definido de vezes, pois ela possui limites fixos.

```
int quantidade = leInteiro("Informe a quantidade de valores");

double total = 0;
for(int contador = 1; contador <= quantidade; contador++) {
    double valor = leReal("Informe o ", contador, "º valor");
    total = total + valor;
}
escrevaL("O valor Total é de R$", total);
```