

Jogo de advinhação

Parte 2

Verificando a resposta do usuário

A cada tentativa, devemos verificar se o usuário acertou ou errou o número

Se ele acertou, devemos parabenizá-lo, caso contrário, dizer que ele errou

Estas mensagens podem ser apresentadas conforme as linhas abaixo

```
printf("\n Parabéns, você acertou. O número secreto era %d \n", numeroSecreto);  
printf("\n O número secreto não é %d \n", palpite);
```

Mas devemos optar entre uma ou outra, acerto ou erro

Verificando a resposta do usuário

Temos que fazer com que o programa tome uma decisão: Qual mensagem mostrar?

Ou seja, qual linha executar?

Em linguagem natural faríamos assim: SE palpite FOR IGUAL a númeroSecreto, ENTÃO parabéns, SENÃO informa o erro

Trata-se de um estrutura condicional, assim representada em C

```
if(palpite==numeroSecreto)
{
    printf("\n Parabéns, você acertou. O número secreto era %d \n", numeroSecreto);
}
else
{
    printf("\n O número secreto não é %d \n", palpite);
}
```

Verificando a resposta do usuário

If é a tradução para inglês de SE

Else é a tradução de SENÃO

O ENTÃO é implícito

```
if (palpite == numeroSecreto)
{
    printf("\n Parabéns, você acertou. O número secreto era %d \n", numeroSecreto);
}
else
{
    printf("\n O número secreto não é %d \n", palpite);
}
```

Dentro do parênteses colocamos uma ou mais condições

== é um operador relacional, ele compara o valor de duas variáveis

O conteúdo dentro dos blocos será executado dependendo do resultado do operador relacional: Verdadeiro/Falso e Então/Senão, respectivamente

Atividade

Pesquise outros operadores relacionais do C e preencha a tabela

Operador	Descrição

Inclua mensagens convidando o usuário a continuar jogando, em caso de acerto, ou tentar novamente, em caso de erro

Dando dicas ao usuário

Considerando que o usuário possa ter errado o número, o jogo deve dar dicas a ele

Ou seja, dizer se o número secreto é maior ou menor que o palpite do usuário

Evidentemente, esta dica deve estar dentro do else de nossa estrutura condicional

```
if (palpite == numeroSecreto)
{
    printf("\n Parabéns, você acertou. O número secreto era %d \n", numeroSecreto);
    printf("\n Jogue outra partida");
}
else
{
    printf("\n O número secreto não é %d \n", palpite);
    printf("\n Tente novamente");
}
```

Dando dicas ao usuário

Teremos que testar se o palpite é maior ou menor que o número secreto

Portanto, precisaremos de outra estrutura condicional dentro do else

Quando colocamos um if dentro de outro, temos o que se chama de estruturas condicionais aninhadas

Faça esta alteração

No próximo slide será apresentada uma solução

OBS: em programação não existem “gabaritos” de código. Há várias soluções possíveis, desde que sejam soluções corretas.

Dando dicas ao usuário

O código abaixo deve ser inserido dentro do else

```
if (palpite > numeroSecreto)
{
    printf("\n O número secreto é menor que %d \n", palpite);
}
else
{
    printf("\n O número secreto é maior que %d \n", palpite);
}
```


Dando dicas ao usuário

Exemplos de execução do jogo

```
*****  
Bem-vindo ao Jogo de Advinhação  
*****  
:)
```

Digite um número:10

O número secreto é maior que 10

Tente novamente

```
*****  
Bem-vindo ao Jogo de Advinhação  
*****  
:)
```

Digite um número:100

O número secreto é menor que 100

Tente novamente

Caminhos do programa

A cada trecho do programa foi atribuído um número

Isto facilitará o entendimento e teste do programa

```
numeroSecreto=30;

1 printf("\n Digite um número:");
  scanf("%d",&palpite);

2 if(palpite==numeroSecreto)
  {
    printf("\n Parabéns, você acertou. O número secreto era %d \n", numeroSecreto);
    printf("\n Jogue outra partida");
  }
  else
  {
3     if(palpite>numeroSecreto)
      {
        printf("\n O número secreto é menor que %d \n", palpite);
      }
      else
      {
4          printf("\n O número secreto é maior que %d \n", palpite);
        }
        printf("\n Tente novamente");
      }
  }
```

Caminhos do programa

Podemos fazer o seguinte grafo com os possíveis caminhos de execução do programa

A trecho 1 sempre será executado

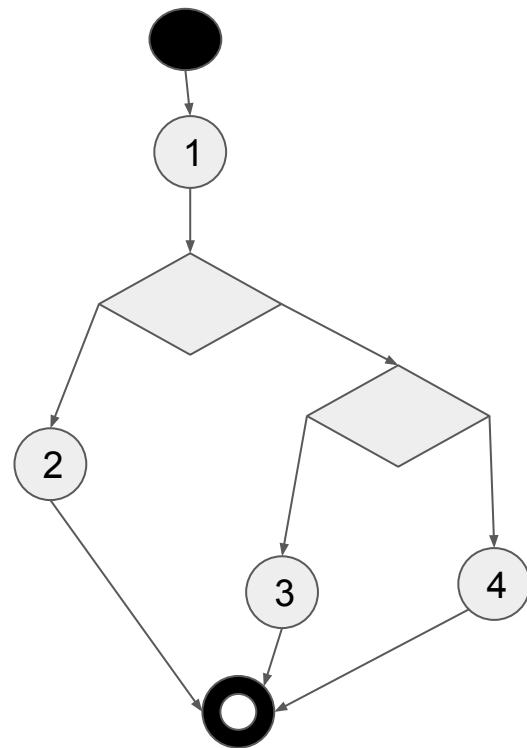
Se o palpite estiver certo, o trecho 2 é executado

Caso contrário, pode ser executado o trecho 3 ou 4

Após os trechos 2, 3 e 4 o programa termina a execução

Utilizando alguma ferramenta de sua preferência...

... reproduza este fluxograma preenchendo as condições



Testando o programa

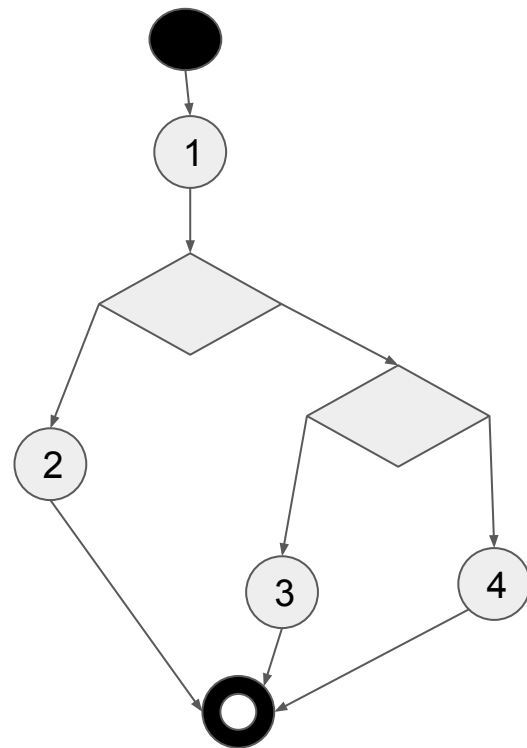
Para cada trecho temos um resultado esperado do programa

Identifique o resultado esperado de cada trecho

Em seguida, indique possíveis valores da variável palpite que permita a execução de cada trecho

Usando estes valores, teste o seu jogo e verifique se ele está apresentando os resultados esperados

Este é um dos princípios mais básicos de Teste de Software



Boas Práticas de Programação

Boas práticas permitem o entendimento do programa

Use nomes de variáveis que indiquem seu conteúdo (evitem x, y, a,b)

Façam indentação (sem indentação é quase impossível descobrir se um trecho de código está dentro ou fora de uma estrutura condicional ou de repetição)

Comentem o programa (isto facilita entender o que cada linha ou trecho de código faz)

Até aqui não comentamos nosso programa. Vamos fazer isso agora