

Descrição

O jogo consiste em adivinhar um número inteiro, entre 1 e 100, gerado aleatoriamente pelo computador. Após iniciar o jogo, o jogador terá 10 chances para acertar o número. O jogo terminará quando o jogador esgotar suas chances ou quando seu palpite igualar-se ao número gerado pelo computador.

A cada insucesso, o palpite do jogador poderá assumir um dos seguintes estados:

Tabela 1

ESTADO	REGRA
MUITO QUENTE (MQ)	Se ele distar até 3 unidades do número gerado pelo computador.
QUENTE (QT)	Se ele distar de 4 a 6 unidades do número gerado pelo computador.
FRIO (FR)	Se ele distar de 7 a 9 unidades do número gerado pelo computador.
MUITO FRIO (MF)	Se ele distar 10 ou mais unidades do número gerado pelo computador.

No primeiro insucesso, o jogador receberá uma das seguintes mensagens, dependendo do estado obtido pelo seu palpite:

Tabela 2

ESTADO	MENSAGEM
MQ	Está muito quente.
QT	Está quente.
FR	Está frio.
MF	Está muito frio.

Daí em diante, o jogador receberá uma das mensagens abaixo, dependendo do estado atual do seu palpite, relativamente ao estado do seu palpite anterior:

Tabela 3

ESTADO ANTERIOR	ESTADO ATUAL			
	MQ	QT	FR	MF
MQ	Ainda continua muito quente.	Oops, deu uma esfriada, mas ainda está quente.	Oops, agora ficou frio.	Oops, agora ficou muito frio.
QT	Oops, deu uma esquentada e agora ficou muito quente.	Ainda continua quente.	Oops, deu uma esfriada e agora ficou frio.	Oops, deu uma baita esfriada e agora ficou muito frio.
FR	Oops, deu uma baita esquentada e agora ficou muito quente.	Oops, deu uma esquentada e agora ficou quente.	Ainda continua frio.	Oops, deu uma esfriada e agora ficou muito frio.
MF	Oops, deu uma baita esquentada e agora ficou muito quente.	Oops, deu uma esquentada e agora ficou quente.	Oops, deu uma esquentada, mas ainda está frio.	Ainda continua muito frio.

No caso do jogador emitir um palpite que se iguale ao número gerado pelo computador, ele receberá uma mensagem de parabéns.

Durante o encerramento do jogo, seja pelo esgotamento das chances de sucesso ou pelo acerto do número gerado pelo computador, o jogador receberá uma mensagem que resume seu desempenho durante a jogada.

Solução por refinamentos sucessivos

Refinamento 0:

```
< gerar número aleatório entre 1 e 100 >
< zerar contador de acertos >
< apresentar jogo ao usuário >
< executar laço de 10 tentativas: >
  < ler (chute) >
  < se acertou: >
    < parabenizar o usuário >
    < incrementar contador de acertos >
    < quebrar laço >
  < senão: >
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status >
< comunicar resultado ao usuário >
```

Refinamento 1:

```
// gerar número aleatório entre 1 e 100
  randomize;
  n := random(100);
< zerar contador de acertos >
< apresentar jogo ao usuário >
< executar laço de 10 tentativas: >
  < ler (chute) >
  < se acertou: >
    < parabenizar o usuário >
    < incrementar contador de acertos >
    < quebrar laço >
  < senão: >
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e spacejar para a próxima tentativa >
< comunicar resultado ao usuário >
```

Refinamento 2:

```
// gerar número aleatório entre 1 e 100
  randomize;
  n := random(100);
// zerar contador de acertos
  acertos := 0;
< apresentar jogo ao usuário >
< executar laço de 10 tentativas: >
  < ler (chute) >
  < se acertou: >
    < parabenizar o usuário >
    < incrementar contador de acertos >
    < quebrar laço >
  < senão: >
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e spacejar para a próxima tentativa >
< comunicar resultado ao usuário >
```

Refinamento 3:

```
// gerar número aleatório entre 1 e 100
  randomize;
  n := random(100);
// zerar contador de acertos
  acertos := 0;
// apresentar jogo ao usuário

  writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

< executar laço de 10 tentativas: >

- < ler (chute) >
- < se acertou: >
 - < parabenizar o usuário >
 - < incrementar contador de acertos >
 - < quebrar laço >
- < senão: >
 - < determinar status >
 - < gerar mensagem correspondente >
 - < alterar status e espaçar para a próxima tentativa >

< comunicar resultado ao usuário >

Refinamento 4:

```
// gerar número aleatório entre 1 e 100
  randomize;
  n := random(100);
// zerar contador de acertos
  acertos := 0;
// apresentar jogo ao usuário

  writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

< executar laço de 10 tentativas: >

```
  for j := 1 to 10 do
  begin
    < ler (chute) >
    < se acertou: >
      < parabenizar o usuário >
      < incrementar contador de acertos >
      < quebrar laço >
    < senão: >
      < determinar status >
      < gerar mensagem correspondente >
      < alterar status e espaçar para a próxima tentativa >
  end;
```

< comunicar resultado ao usuário >

Refinamento 5:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  < se acertou: >
    < parabenizar o usuário >
    < incrementar contador de acertos >
    < quebrar laço >
  < senão: >
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e espacejar para a próxima tentativa >
end;
```

< comunicar resultado ao usuário >

Refinamento 6:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    < parabenizar o usuário >
    < incrementar contador de acertos >
    < quebrar laço >
  end
  // senão:
  else
  begin
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e espacejar para a próxima tentativa >
  end;
end;
< comunicar resultado ao usuário >
```

Refinamento 7:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
      begin
        writeln('Valor invalido!');
        write ('Tente de novo: ');
      end;
  until (erro = 0);
  // se acertou:
  if chute = n then
    begin
      // parabenizar o usuário
      writeln;
      writeln('Parabens !');
      < incrementar contador de acertos >
      < quebrar laço >
    end
  // senão:
  else
    begin
      < determinar status >
      < gerar mensagem correspondente >
      < alterar status e espacelar para a próxima tentativa >
    end;
  end;
end;
< comunicar resultado ao usuário >
```

Refinamento 8:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
      begin
        writeln('Valor invalido!');
        write ('Tente de novo: ');
      end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    < quebrar laço >
  end
  // senão:
  else
  begin
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e espacejar para a próxima tentativa >
  end;
end;
< comunicar resultado ao usuário >
```

Refinamento 9:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    // quebrar laço
    break;
  end
  // senão:
  else
  begin
    < determinar status >
    < gerar mensagem correspondente >
    < alterar status e espacejar para a próxima tentativa >
  end;
end;
< comunicar resultado ao usuário >
```


Refinamento 10:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    // quebrar laço
    break;
  end
  // senão:
  else
  begin
    // determinar status
    if j = 1 then
    begin
      e1 := status(n, chute); // determinar o status do primeiro palpite
      e2 := '';              // anular o status do próximo palpite
    end
    else
    begin
      e2 := status(n, chute); // determinar o status dos demais palpites
      < gerar mensagem correspondente >
      < alterar status e espacelar para a próxima tentativa >
    end;
  end;
end;
end;
end;
< comunicar resultado ao usuário >
```

Refinamento 11:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    // quebrar laço
    break;
  end
  // senão:
  else
  begin
    // determinar status
    if j = 1 then
    begin
      e1 := status(n, chute); // determinar o status do primeiro palpite
      e2 := '';              // anular o status do próximo palpite
    end
    else
    begin
      e2 := status(n, chute); // determinar o status dos demais palpites
      // gerar mensagem correspondente
      emiteMensagem(e1, e2);
      < alterar status e espacelar para a próxima tentativa >
    end;
  end;
end;
end;
end;
< comunicar resultado ao usuário >
```

Refinamento 12:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    // quebrar laço
    break;
  end
  // senão:
  else
  begin
    // determinar status
    if j = 1 then
    begin
      e1 := status(n, chute); // determinar o status do primeiro palpite
      e2 := '';              // anular o status do próximo palpite
    end
    else
    begin
      e2 := status(n, chute); // determinar o status dos demais palpites
      // gerar mensagem correspondente
      emiteMensagem(e1, e2);
      // alterar status e espacejar para a próxima tentativa
      if e2 <> '' then
        e1 := e2;
      writeln;
    end;
  end;
end;
end;
< comunicar resultado ao usuário >
```

Refinamento 13:

```
// gerar número aleatório entre 1 e 100
randomize;
n := random(100);
// zerar contador de acertos
acertos := 0;
// apresentar jogo ao usuário

writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');

writeln;

```
// executar laço de 10 tentativas:
for j := 1 to 10 do
begin
  // ler (chute)
  write ('Tentativa ', j, ': ');
  repeat
    readln(resp);
    val(resp, chute, erro);
    if erro <> 0 then
    begin
      writeln('Valor invalido!');
      write ('Tente de novo: ');
    end;
  until (erro = 0);
  // se acertou:
  if chute = n then
  begin
    // parabenizar o usuário
    writeln;
    writeln('Parabens !');
    // incrementar contador de acertos
    inc(acertos);
    // quebrar laço
    break;
  end
  // senão:
  else
  begin
    // determinar status
    if j = 1 then
    begin
      e1 := Status(n, chute); // determinar o status do primeiro palpite
      e2 := '';              // anular o status do próximo palpite
    end
    else
    begin
      e2 := Status(n, chute); // determinar o status dos demais palpites
      // gerar mensagem correspondente
      emiteMensagem(e1, e2);
      // alterar status e espacejar para a próxima tentativa
      if e2 <> '' then
        e1 := e2;
      writeln;
    end;
  end;
end;
end;
```

```
// comunicar resultado ao usuário
writeln;
if acertos = 0 then
    writeln('Infelizmente, apos 10 tentativas, voce nao conseguiu acertar o numero:',
n, '.')
else
    writeln('Voce acertou o numero ', n, ' apos ', j, ' tentativas.');
```

A função Status

A função **Status** toma como parâmetros o número gerado pelo computador e o palpite formulado pelo jogador. Com base nesses valores, ela retorna o status desse palpite, de acordo com as regras estabelecidas na Tabela 1.

```
function Status(n: integer; chute: integer) : string;
var temp: integer;
begin
    temp := abs(chute - n);
    if temp <= 3 then           Status := 'MQ'
    else if temp <= 6 then       Status := 'QT'
        else if temp <= 9 then   Status := 'FR'
            else                 Status := 'MF';
end;
```

O procedimento emiteMensagem

O procedimento **emiteMensagem** toma como parâmetros os estados dos palpites anterior e atual do jogador e emite uma das mensagens definidas na Tabela 2, se for o primeiro insucesso do jogador, ou uma das mensagens correspondente à combinação desses estados, de acordo com a transições definidas nas Tabela 3.

```
procedure emiteMensagem(Var e1, e2: string);
begin
    case e1 of
        'MQ': case e2 of
            'MQ': writeln('Ainda continua muito quente.');
            'QT': writeln('Oops, deu uma esfriada, mas ainda estah quente.');
            'FR': writeln('Oops, agora ficou frio.');
            'MF': writeln('Oops, agora ficou muito frio.');
            else writeln('Estah muito quente.');
        end;
        'QT': case e2 of
            'MQ': writeln('Oops, deu uma esquentada e agora ficou muito quente.');
            'QT': writeln('Ainda continua quente.');
            'FR': writeln('Oops, deu uma esfriada e agora ficou frio.');
            'MF': writeln('Oops, deu uma baita esfriada e agora ficou muito frio.');
            else writeln('Estah quente.');
        end;
        'FR': case e2 of
            'MQ': writeln('Oops, deu uma baita esquentada e agora ficou muito quente.');
            'QT': writeln('Oops, deu uma esquentada e agora ficou quente.');
            'FR': writeln('Ainda continua frio.');
            'MF': writeln('Oops, deu uma esfriada e agora ficou muito frio.');
            else writeln('Estah frio.');
        end;
        'MF': case e2 of
            'MQ': writeln('Oops, deu uma baita esquentada e agora ficou muito quente.');
            'QT': writeln('Oops, deu uma esquentada e agora ficou quente.');
            'FR': writeln('Oops, deu uma esquentada, mas ainda estah frio.');
            'MF': writeln('Ainda continua muito frio.');
            else writeln('Estah muito frio.');
        end;
    end;
end;
```

O programa PASCAL

```
program guessgame;

function Status(n: integer; chute: integer) : string;
begin
    if chute < (n-9) then
        status := 'MF'
    else
        if chute < (n-6) then
            status := 'FR'
        else
            if chute < (n-3) then
                status := 'QT'
            else
                if chute <= (n+3) then
                    status := 'MQ'
                else
                    if chute <= (n+6) then
                        status := 'QT'
                    else
                        if chute <= (n+9) then
                            status := 'FR'
                        else
                            status := 'MF';
end;

procedure emiteMensagem(var e1, e2: string);
begin
    case e1 of
        'MQ': case e2 of
            'MQ': writeln('Ainda continua muito quente.');
```

```
'MF': case e2 of
    'MQ': writeln('Oops, deu uma baita esquentada e agora ficou muito
quente.');
```

```
    'QT': writeln('Oops, deu uma esquentada e agora ficou quente.');
```

```
    'FR': writeln('Oops, deu uma esquentada, mas ainda estah frio.');
```

```
    'MF': writeln('Ainda continua muito frio.');
```

```
    else writeln('Estah muito frio.');
```

```
end;
```

```
end;
```

```
end;
```

```
var
    acertos, chute, n, j, erro: integer;
    e1, e2, resp: string;
```

```
begin
    // gerar número aleatório entre 1 e 100
    randomize;
    n := random(100);
    // zerar contador de acertos
    acertos := 0;
    // apresentar jogo ao usuário
    writeln('Voce tem 10 chances de acertar o numero que eu estou pensando.');
```

```
    writeln('Trata-se de um valor entre 1 e 100. Entao vamos la.');
```

```
    writeln;
```

```
    // executar laço de 10 tentativas:
```

```
    for j := 1 to 10 do
    begin
        // < ler (chute)
        write ('Tentativa ', j, ': ');
        repeat
            readln(resp);
            val(resp, chute, erro);
            if erro <> 0 then
                begin
                    writeln('Valor invalido!');
```

```
                    write ('Tente de novo: ');
```

```
                end;
```

```
            until (erro = 0);
```

```
            // se acertou:
```

```
            if chute = n then
            begin
                // parabenizar o usuário
                writeln;
```

```
                writeln('Parabens !');
```

```
                // incrementar contador de acertos
                inc(acertos);
```

```
                // quebrar laço
                break;
```

```
            end
```

```
            // senão:
```

```
            else
            begin
                // determinar status
```

```
    if j = 1 then                                // determinar estado1
    begin
        e1 := status(n, chute);
        e2 := '';
    end
    else                                          // determinar estado2
        e2 := status(n, chute);
        // gerar mensagem correspondente
        emiteMensagem(e1, e2);
        // alterar status
        if e2 <> '' then
            e1 := e2;
        writeln;
    end;
end;
// comunicar resultado ao usuário
writeln;
if acertos = 0 then
    writeln('Infelizmente, apos 10 tentativas, voce nao conseguiu acertar o
numero ', n, '.')
else
    writeln('Voce acertou o numero ', n, ' apos ', j, ' tentativas.');
```

readln;

end.

O Enunciado do Trabalho

Após estudar e executar este programa, você notará algumas limitações do seu projeto:

- Em primeiro lugar, o programa não “memoriza” os chutes dados pelo jogador, o que permite que ele repita o mesmo chute, até mesmo 10 vezes, se assim ele o desejar.
- Sempre que o estado 1 e 2 são iguais, o programa emite uma mensagem informando ao jogador que o estado do seu chute continua o mesmo, sem dar-lhe qualquer pista se ele se aproximou ou se afastou do número gerado pelo computador.
- Da forma como o programa foi projetado, todo chute cuja distância para o número gerado é maior ou igual a 10 unidades é considerado muito frio. Entretanto, um chute que diste 10 unidades está, evidentemente, menos frio do que um que diste, por exemplo, 50 unidades. Analogamente, qualquer chute que diste até 3 unidades do número gerado pelo computador está muito quente; porém, um chute que diste apenas uma unidade está muito mais quente do que um que diste 3 unidades. Isso vale também para os demais estados.

Nosso trabalho consiste em modificar este programa para implementar as seguintes modificações em seu projeto original:

- Sempre que o jogador repetir um chute que ele já tenha dado anteriormente, o programa deverá recusá-lo e emitir uma mensagem informando ao jogador sobre essa situação e pedindo para ele tentar outro chute. Obviamente, esse chute duplicata não deverá ser contabilizado com uma das 10 tentativas que o jogador tem direito. (SUGESTÃO: Utilizar um array de 10 elementos para armazenar os chutes do jogador.)
- Sempre que o estado 1 e 2 forem iguais, o programa deverá emitir uma mensagem informando ao jogador que o estado do seu chute continua o mesmo, mas que deu uma esquentada ou uma esfriada em relação ao palpite anterior. (SUGESTÃO: Comparar a distância do chute atual com a distância do chute anterior - se for maior, esfriou; caso contrário, esquentou.)
- A fim de melhorar a orientação do jogador, você deverá criar mais três possíveis estados para cada palpite: 'Fervendo' (FV), 'Morno' (MN) e 'Congelando' (CG), obedecendo a seguinte ordem: FV - MQ - QT - MO -FR - MF - CG. Cabe a você definir os intervalos correspondentes a cada um desses estados, os quais não precisam guardar qualquer relação com os intervalos utilizados na presente versão do programa.

NOTA: Para realizar este trabalho, você precisará estudar o Capítulo 5 (Estruturas de decisão ou seleção) e o Capítulo 9 (Sub-Rotinas) da apostila de Pascal que disponibilizei para a turma, em nossa área do dropbox, na pasta Material Didático\Apostilas, bem como assistir as videoaulas 11 a 15, disponibilizadas no mesmo espaço, na pasta Material Didático\Videoaulas. Além disso, sugiro que vocês deem uma olhada nos exemplos de uso de funções e procedimentos no ambiente do Lazarus, que também estão disponíveis nessa mesma área, na pasta Exemplos\Subprogramas.

BOM TRABALHO !