

O Misterioso Número “6174”

Mário Leite

...

Dentro da Matemática existem vários ramos de estudos e pesquisa, mas, a “Teoria dos Números” é a que mais gosto, pois, traduz melhor a sua essência de ciência exata. Este ramo da Matemática apresenta assuntos muito interessantes; e mesmo que não tenham uma aplicação imediata para os programadores é fonte de intensas atividades intelectuais. E isto é muito importante para os iniciantes em programação, pois apresenta casos desafiadores durante o aprendizado, indo além do clássico “*média das notas de um aluno*”; e um desses casos é o misterioso número **6174**. Este número é realmente misterioso, tornando-se alvo de intensas pesquisas de interesse dos matemáticos, a partir dos trabalhos do indiano *Dattathreya Ramachandra Kaprekar* (1905-1986). Esse matemático descobriu que o número 6174 pode ser criado a partir de outro de quatro dígitos, desde que, pelo menos, dois deles não sejam iguais. Assim, descobriu-se que 6174 tem algumas características interessantes:

- 1) Pode ser escrito como a soma das três primeiras potências de 18 ($18 + 18^2 + 18^3$).
- 2) A soma dos quadrados dos seus fatores primos é um quadrado exato.
- 3) É um *número 7-suave* (nenhum de seus fatores primos é superior a 7).
- 4) É um *número Harshad* (divisível pela soma de seus dígitos).

A **figuras 1** mostram exemplos de saídas do programa “**Numero6174**”, em *Pascal*. Para os programadores de outras linguagens de programação, eles podem se basear no Algoritmo para codificar o programa, como fiz, adicionalmente, em *SciLab*, cuja saída é mostrada na **figura 2**. Como exercício, deixo para os programadores em geral, comprovarem que a partir do momento que aparecer 6174, não adianta continuar, pois o resultado SEMPRE será **6174**; uma constante: que foi, justamente, a descoberta do citado matemático indiano!

E, especificamente, para o pessoal de “Ciência da Computação”: criar um programa que comprove que o máximo de iterações para encontrar 6174, a partir de um número de quatro dígitos, dentro das especificações exigidas, é **sete**, como pode ser confirmado na **figura 2**.

Dá um bom TCC, pessoal; mãos à obra!

Algoritmo:

Inicio

- 1) **Pegue** um número N de quatro dígitos.
- 2) **Se**(não tiver pelo menos dois dígitos diferentes tamanho 4) **Então**
 Volte ao Passo 1
 FimSe
- 3) **Crie** um número X como resultado da classificação de N em ordem decrescente.
- 4) **Crie** outro número Y com os dígitos do número X em ordem crescente.
- 5) **Se**(Y tiver menos que quatro dígitos) **Então**
 Acrescente zeros à direita de Y
 FimSe
- 6) **Subtraia** X de Y
- 7) **Se**(o resultado da subtração for diferente de 6174) **Então**
 Faça N igual ao resultado da subtração (com quatro dígitos)
 Volte ao Passo 3
 FimSe
- 8) **Mostre** o resultado da subtração

Fim

```
Program Numero6174;
//Em Pascal
//Autor: Mário Leite
//-----
var VetRep, VetNum1, VetNum2: array[1..4] of string;
    dig1, dig2, numOrig, numCres, numDecres: string;
    i, j, itera, erro, sub, tamNum, numMaior, numMenor: integer;
    Aux: char; repetiu, cond, Achou: boolean;
Begin
  Repeat //leitura e validação do número lido
    Write('Digite um número de quatro dígitos diferentes: ');
    ReadLn(numOrig);
    tamNum := Length(numOrig);
    repetiu := False;
    //Verifica se há repetições
    repetiu := False;
    dig1 := numOrig[1];
    cond := ((numOrig[2]=dig1) and (numOrig[3]=dig1) and (numOrig[4]=dig1));
    if(cond) then begin
      repetiu := True;
    end;
  Until((tamNum=4) and (not(repetiu))); //fim da leitura/validação
  //-----
  //Rotina para obter o número 6174
  Achou := False;
  itera := 0;
  while (not(Achou)) do begin
    //Cria o número em ordem decrescente
    for i:=1 to 3 do begin
      for j:=(i+1) to 4 do begin
        if(numOrig[i] < numOrig[j]) then begin
          Aux := numOrig[i];
          numOrig[i] := numOrig[j];
          numOrig[j] := Aux;
        end;
      end;
    end;
    numDecres := numOrig;
    //Cria o número em ordem crescente
```

```

numCres := '';
for i:=4 downto 1 do begin
    numCres := numCres + numDecres[i];
end;
if (Length(numCres)<4) then begin
    numCres := '0' + numCres;
end;
//Faz a subtração e verifica se resulta em 6174
Val(numDecres, numMaior, erro);
Val(numCres, numMenor, erro);
sub := numMaior - numMenor;
itera := itera + 1;
WriteLn(sub);
if (sub=6174) then
    begin
        Achou := True;
        break; //terminou: abandona o loop
    end
else begin //gera novo número para trabalhar
    Str(sub, numOrig);
    if (Length(numOrig)<4) then begin
        numOrig := '0' + numOrig;
    end;
end;
end; //fim do loop de criação do número 6174
WriteLn('');
WriteLn('Número encontrado após ', itera, ' iterações: ', sub);
ReadKey;
End. //-----

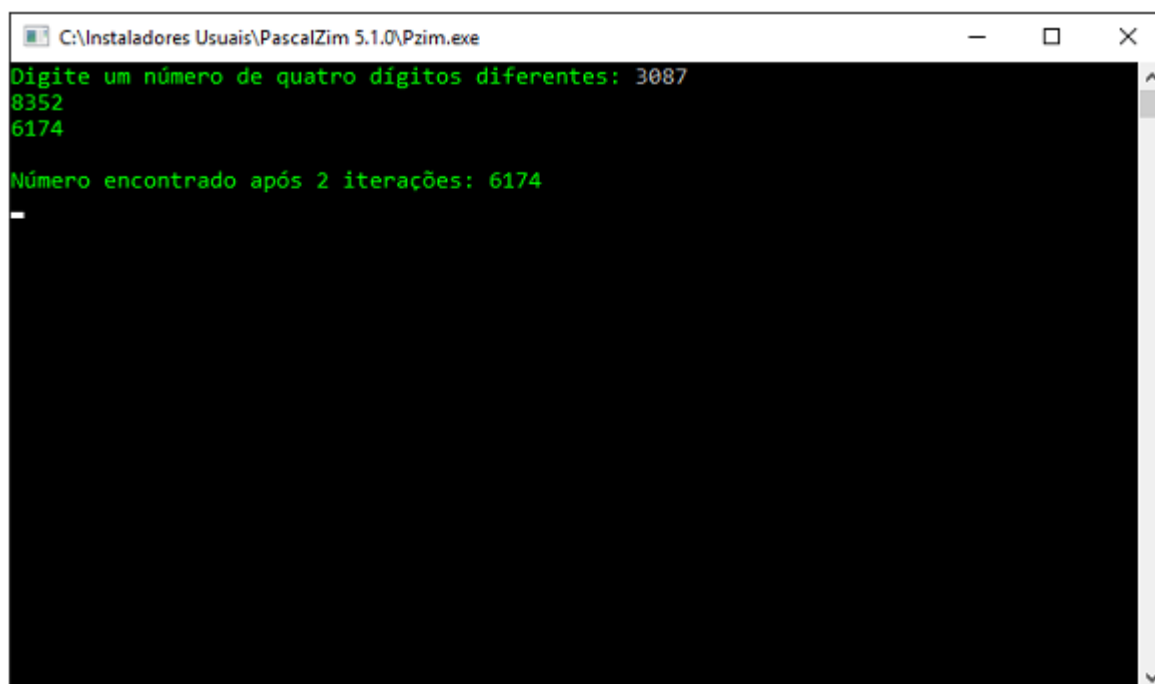
//Programa "Numero6174"
//Em SciLab
//Autor: Mário Leite
//Inicio -----
clc //limpa a tela
tamNum = 3
repetiu = %t
VetNum1 = []
//Leitura e validação do número lido
while ((tamNum<4) & (repetiu)) do
    numOrig=input("Digite um número de quatro dígitos diferentes:", "string")
    tamNum = length(numOrig)
    //Cria vetor com os dígitos de numOrig
    VetNum1(1) = part(numOrig,1)
    VetNum1(2) = part(numOrig,2)
    VetNum1(3) = part(numOrig,3)
    VetNum1(4) = part(numOrig,4)
    //Verifica se há repetições
    repetiu = %f
    dig1 = VetNum1(1)
    cond1 = ((VetNum1(2)==dig1) & (VetNum1(3)==dig1) & (VetNum1(4)==dig1))
    if (cond1) then
        repetiu = %t
    end;
end; //fim da leitura/validação do número lido
//-----
//Rotina para obter o número 6174
Achou = %f
itera = 0
while (~(Achou)) do
    //Cria vetor com os dígitos de numOrig
    VetNum1(1) = part(numOrig,1)
    VetNum1(2) = part(numOrig,2)
    VetNum1(3) = part(numOrig,3)

```

```

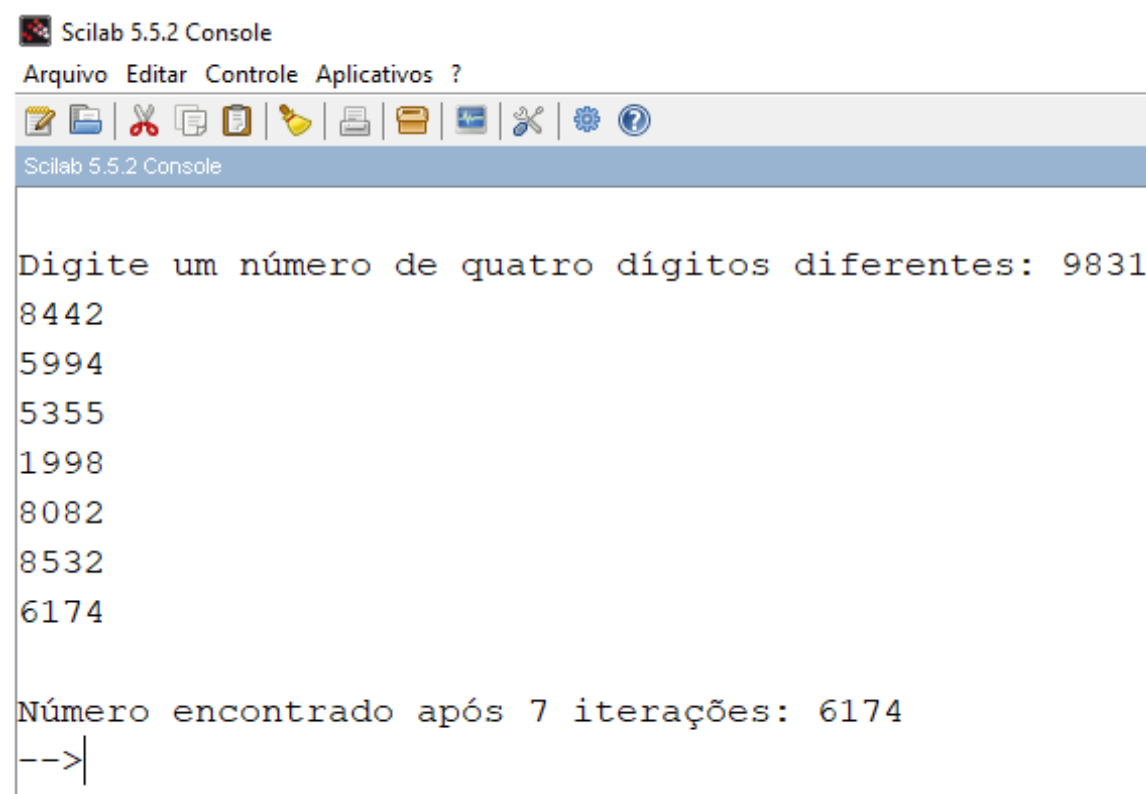
VetNum1(4) = part(numOrig,4)
//Classifica o número lido em ordem decrescente
for i=1:3 do
    for j=(i+1):4 do
        digi = eval(VetNum1(i))
        digj = eval(VetNum1(j))
        if(digi < digj) then
            Aux = VetNum1(i)
            VetNum1(i) = VetNum1(j)
            VetNum1(j) = Aux
        end;
    end;
end;
//Cria o número em ordem decrescente
numDecres = VetNum1(1) + VetNum1(2) + VetNum1(3) + VetNum1(4)
//Cria o número em ordem crescente
numCres = VetNum1(4) + VetNum1(3) + VetNum1(2) + VetNum1(1)
//Faz a subtração e verifica se resulta em 6174
numMaior = eval(numDecres)
numMenor = eval(numCres)
sub = numMaior - numMenor;
if(sub<1000) then
    subs = "0" + string(sub)
    printf("%s \n",subs)
else
    printf("%d \n",sub)
end;
itera = itera + 1 //incrementa o número de iterações
if(sub==6174) then
    Achou = %t
    break //terminou: abandona o loop
else //gera novo número para trabalhar
    if(sub<1000) then
        numOrig = subs
    else
        numOrig = string(sub)
    end;
end;
end; //fim do loop de criação do número 6174
printf("\n")
printf("%s %d %s %d","Número encontrado após ",itera,"iterações: ", sub)
//Fim -----

```



```
C:\Instaladores Usuais\PascalZim 5.1.0\Pzim.exe
Digite um número de quatro dígitos diferentes: 3087
8352
6174
Número encontrado após 2 iterações: 6174
```

Figura 1 - Exemplo de saída do programa em Pascal



```
Scilab 5.5.2 Console
Arquivo Editar Controle Aplicativos ?
Scilab 5.5.2 Console

Digite um número de quatro dígitos diferentes: 9831
8442
5994
5355
1998
8082
8532
6174

Número encontrado após 7 iterações: 6174
-->
```

Figura 2 - Exemplo de saída do programa em SciLab