

Verificando Dados Repetidos

Mário Leite

...

Uma das ações muito frequentes é a necessidade de verificar se existem dados repetidos nas entradas, tal como acontece com as dezenas sorteadas nos concursos da Mega Sena. Mas, neste caso não há necessidade de verificar se alguma dezena sorteada se repetiu, pois, uma vez sorteada não há possibilidade dessa dezena sair novamente porque cada sorteio é feito sobre exatamente sessenta dezenas, com bolas numeradas de 01 a 60 (sem repetição). Por outro lado, no caso de entrada de dados sequencialmente digitados pode ser que o usuário digite o mesmo dado (no caso um valor numérico) mais de uma vez. Assim, no caso de um programa para gerar as seis dezenas para a Mega Sena, existe a possibilidade de sair dezenas repetidas; então, neste caso, é necessário verificar se houve alguma repetição. Uma solução seria armazenar os elementos num vetor e em seguida verificar se o elemento i gerado e o $i-1$ são iguais; se forem iguais um desses elementos deve ser eliminado; e a verificação deve ser feita para todos elementos subsequentes.

A função **VerificaRepeticao()** atende a este requisito: ela recebe um vetor de inteiros e faz a verificação dos elementos desse vetor. retorna **Verdadeiro (V)** se existe alguma repetição ou **Falso (F)** se não existem repetições.

Felizmente, existe uma solução mais elegante e mais eficiente para resolver este problema: verificar repetições já durante a digitação e não após a geração de todos os elementos; e é isto que faz o programa **VetorSemfRepeticao**: ele cria um vetor de inteiros e sem repetições à medida que o usuário vai digitando os números. E tanto a função quanto o programa, foram planejados em pseudocódigo (adaptáveis para quaisquer linguagens de programação).

```
Função VerificaRepeticao (VetNum[ ], x:inteiro) : lógico
//Verifica as entradas dos elementos de um vetor de inteiros.
//Se houver alguma repetição retorna Verdadeiro, caso contrário retorna Falso.
//X é o tamanho do vetor VetNum[ ].
//-----
  Declare m, n, p, y: inteiro
  Repetiu: lógico
Início
  Repetiu ← F
  n ← 1
  Enquanto (n<=x) Faça //início do loop de verificação
    Se (x<2) Então
      Repetiu ← F
    Senão
      Para m De 1 Até (n-1) Faça
        p ← m + 1
        Para y De p Até n Faça
          Se (VetNum[m]=VetNum[n]) Então //repetiu!!
            Repetiu ← V
          FimSe
        FimPara
      FimPara
    FimSe
    n ← n + 1
  FimEnquanto //fim do loop de verificação
  Retorne Repetiu
FimFunção
```

Programa "VetorSemRepeticao"

//Cria dinamicamente um vetor de números inteiros digitados, sem elementos repetidos.

```
//-----  
    Const MAXTAM=100  
    Declare VetNum: array[MAXTAM] de inteiro  
        i, j, k, m, N: inteiro  
Início  
    Repita  
        Escreva("Digite o número de elementos do vetor [min 2", " - max", MAXTAM, "]: ")  
        Leia(N)  
        N ← Int(Abs(N))  
    AtéQue (N>=2) e (N<=MAXTAM)  
    j ← 1  
    Enquanto (j<=N) Faça  
        Escreva("Digite um número: ")  
        Leia(VetNum[j])  
        Se(j>1) Então  
            Para i De 1 Até (j-1) Faça  
                k ← i + 1  
                Para m De k Até j Faça  
                    Enquanto (VetNum[i]=VetNum[j]) Faça //repetiu!!  
                        {Tenta um novo número para este elemento}  
                        Escreva("Repetido! Tente outro número: ")  
                        Leia(VetNum[j])  
                    FimEnquanto  
                FimPara  
            FimSe  
            j ← j + 1 //próximo elemento do vetor  
        FimEnquanto  
    LimpaTela  
    Para i De 1 Até N Faça  
        Escreva(VetNum[i], " ")  
    FimPara  
FimPrograma
```