

Laços de Repetição

Para

- E se houver um problema em que sejam necessárias um número determinado de repetições?

Para

- E se houver um problema em que sejam necessárias um número determinado de repetições?
- Por exemplo, se quiséssemos pedir ao usuário que digitasse 10 valores. Poderíamos utilizar a instrução Leia repetidas vezes. Porém se ao invés de 10 valores precisássemos de 100, essa tarefa se tornaria muito extensa. Para resolver problemas como esse, podemos usar um laço de repetição com variável de controle. No português, ele é conhecido como **para**.

Para

- E se houver um problema em que sejam necessárias um número determinado de repetições?
- Por exemplo, se quiséssemos pedir ao usuário que digitasse 10 valores. Poderíamos utilizar a instrução Leia repetidas vezes. Porém se ao invés de 10 valores precisássemos de 100, essa tarefa se tornaria muito extensa. Para resolver problemas como esse, podemos usar um laço de repetição com variável de controle. No português, ele é conhecido como **para**.
- O laço de repetição com variável de controle facilita a construção de algoritmos com número definido de repetições, pois possui um contador (variável de controle) embutido no comando com o incremento automático.

Para

- Então, necessita de um número determinado de repetições utilizando um contador. O contador deve ser inicializado antes do laço e deve ser incrementado dentro do laço.

Para

- Então, necessita de um número determinado de repetições utilizando um contador. O contador deve ser inicializado antes do laço e deve ser incrementado dentro do laço.
- O laço com variável de controle possui três partes. A inicialização da variável contadora, a definição do valor final do contador e a definição do incremento. Estas três partes são escritas juntas, no início do laço.

Para

- Então, necessita de um número determinado de repetições utilizando um contador. O contador deve ser inicializado antes do laço e deve ser incrementado dentro do laço.
- O laço com variável de controle possui três partes. A inicialização da variável contadora, a definição do valor final do contador e a definição do incremento. Estas três partes são escritas juntas, no início do laço.
- A sintaxe é respectivamente a palavra reservada **para**, abre parênteses, a declaração de uma variável de controle, ponto e vírgula, a condição a ser testada, ponto e vírgula, uma alteração na variável de controle a ser feita a cada iteração, fecha parênteses, e entre chaves as instruções do programa.

Para

Exemplo de Sintaxe

```
01. para (inteiro i = 0; i < 8; i++)  
02. {  
03.     //Codigo a ser executado enquanto a condição for satisfeita.  
04. }
```


Exercício (Para)

- Solicite um número inteiro ao usuário e retorne para ele a tabuada do número informado.

Exemplo de Sintaxe

```
01. para (inteiro i = 0; i < 8; i++)  
02. {  
03.     //Codigo a ser executado enquanto a condição for satisfeita.  
04. }
```

Prática

1. Escreva um programa em Portugol que lê 15 valores inteiros positivos, encontra o maior e o menor deles e mostra o resultado.
2. Faça um programa que verifique e mostre os números entre 1.000 e 2.000 (inclusive) que, quando divididos por 11, produzam o resto igual a 2.
3. Faça um programa que leia um valor n , inteiro e positivo, calcule e mostre a seguinte soma: $S = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$.

Vetores

Vetores

- Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?

Vetores

- Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?
- Seria necessário criar uma variável para cada aluno?

Vetores

- Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?
- Seria necessário criar uma variável para cada aluno?
- E se cada turma tiver quantidade de alunos variáveis?

Vetores

- Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?
- Seria necessário criar uma variável para cada aluno?
- E se cada turma tiver quantidade de alunos variáveis?
- E os nomes de cada um dos alunos?

Vetores

- Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?
- Seria necessário criar uma variável para cada aluno?
- E se cada turma tiver quantidade de alunos variáveis?
- E os nomes de cada um dos alunos?
- Poderíamos armazenar estes dados em variáveis, porém o controle de muitas variáveis em um programa não é uma solução prática.

Vetores

- Ao invés disso, utiliza-se uma estrutura de dados que agrupa todos estes valores em um único nome. Esta estrutura chama-se vetor.

Vetores

- Ao invés disso, utiliza-se uma estrutura de dados que agrupa todos estes valores em um único nome. Esta estrutura chama-se vetor.
- Um vetor pode ser visto como uma variável que possui diversas posições, e com isso armazenar diversos valores, porém todos do mesmo tipo.

Vetores

- Ao invés disso, utiliza-se uma estrutura de dados que agrupa todos estes valores em um único nome. Esta estrutura chama-se vetor.
- Um vetor pode ser visto como uma variável que possui diversas posições, e com isso armazenar diversos valores, porém todos do mesmo tipo.
- Assim como as variáveis, o vetor tem que ser declarado. Sua declaração é similar à declaração de variáveis, definindo primeiro o seu tipo, em seguida do seu nome e por fim a sua dimensão entre colchetes (opcional se for atribuir valores a ele na declaração).

Vetores

Exemplo de Sintaxe

```
01. inteiro vetor[5]
02. caracter vetor2[200]
03.
04. //vetores inicializados
05. real vetor3[2] = {1.4,2.5}
06. logico vetor4[4] = {verdadeiro,falso,verdadeiro,verdadeiro}
07. cadeia vetor5[] = {"Questão","Fundamental"}
08.
09. //Mudando o valor do vetor5 na posição 0 de "Questão" para "Pergunta"
10. vetor5[0] = "Pergunta"
```

Vetores

- Elementos individuais são acessados por sua posição no vetor. Como um vetor tem mais de uma posição, deve-se indicar qual posição do vetor se quer fazer acesso. Para isso é necessário usarmos um índice.

Vetores

Exemplo

```
01. programa
02. {
03.     funcao inicio()
04.     {
05.         //Declaração de um vetor de inteiros
06.         // de cinco posições já inicializado.
07.         inteiro vetor[5] = {15,22,8,10,11}
08.
09.         //Imprime o valor 15 correspondente
10.         // ao primeiro elemento do vetor.
11.         escreva(vetor[0])
12.         escreva("\n")
13.
14.         //Imprime o segundo elemento do vetor
15.         escreva(vetor[1])
```

Vetores

- Solicite que o usuário escreva 10 valores, você deve armazenar em um Vetor e depois exibir esses valores somados.
- Solicite que o usuário escreva suas notas, você deve armazenar em um Vetor e depois exibir a média. Dica: pergunte quantas notas ele pretende digitar para você criar o vetor no tamanho correto.
- Solicite que o usuário digite 15 números, você deve armazenar em um vetor e depois exibir o maior e o menor valor.

Vetores

- Escreva um algoritmo que solicite ao usuário a entrada de 5 nomes, e que exiba a lista desses nomes na tela. Após exibir essa lista, o programa deve mostrar também os nomes na ordem inversa em que o usuário os digitou, um por linha.
- Crie um programa que solicite a entrada de 10 números pelo usuário, armazenando-os em um vetor, e então monte outro vetor com os valores do primeiro multiplicados por 5. Exiba os valores dos dois vetores na tela, simultaneamente, em duas colunas (um em cada coluna), uma posição por linha.
- Crie um programa que armazene 10 números digitados pelo usuário em dois vetores: um somente para números pares, e outro somente para números ímpares. Após, exiba os valores dos dois vetores na tela, em sequência. Obs.: As posições que não receberem valores exibirão o número zero. Não se preocupe com isso por enquanto.

Vetores

- Modifique o programa anterior para não aceitar a entrada do número zero, e requisitar a digitação de outro número neste caso.
- Modifique novamente o programa anterior, de modo a não exibir na saída os números zero que são mostrados para todas as posições que não receberam nenhum valor durante a atribuição (e portanto estão vazias).



Siga o Senac em Minas nas Redes Sociais:

