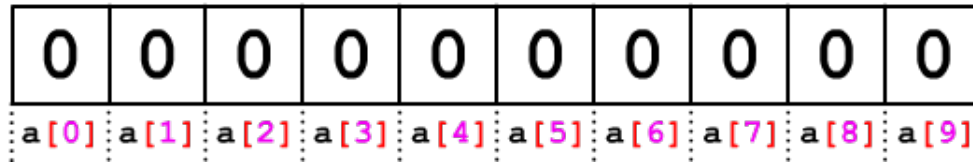


Figura 4.1: Indexação dos Arrays

```
int a[10] = { 0 };
```



Fonte: Elaborada pelo autor

4.2 Exercícios

Exercício 4.1: Escreva um programa que preencha um array de números inteiros de 5 posições a partir de números fornecidos pelo usuário. O programa deve armazenar em um segundo array o cubo de cada elemento do primeiro array. Por fim, o programa deve exibir os valores do array que contém o cubo do primeiro array.

Arquivo com a solução: [ex4.1.c](#)

Entrada

```
array[0] : 4  
array[1] : 5  
array[2] : 7  
array[3] : 10  
array[4] : -8
```

Saída

```
arrayCubo[0] = 64  
arrayCubo[1] = 125  
arrayCubo[2] = 343  
arrayCubo[3] = 1000  
arrayCubo[4] = -512
```

Exercício 4.2: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. Logo em seguida, o programa deve pedir o valor de um número inteiro. O programa deve multiplicar cada um dos valores do array inicial pelo valor fornecido e armazenar, em um segundo array, também de 5 posições, o valor da multiplicação de cada posição do array inicial pelo valor fornecido após a leitura do primeiro array. Por fim, o programa deve exibir os valores do array

que contém a multiplicação de cada item.

Arquivo com a solução: [ex4.2.c](#)

Entrada

```
array[0] : 4  
array[1] : 5  
array[2] : 7  
array[3] : 10  
array[4] : -8  
Multiplicar por: 5
```

Saída

```
arrayMult[0] = 20  
arrayMult[1] = 25  
arrayMult[2] = 35  
arrayMult[3] = 50  
arrayMult[4] = -40
```

Exercício 4.3: Escreva um programa que preencha um array de números decimais de 5 posições com valores fornecidos pelo usuário. Após o preenchimento, o programa deve percorrer o array com os dados fornecidos, calculando o somatório e o produto dos valores contidos no mesmo. Esses resultados devem ser exibidos ao final da execução do programa e devem estar formatados usando duas casas decimais de precisão.

Arquivo com a solução: [ex4.3.c](#)

Entrada

```
array[0] : 4  
array[1] : 5.5  
array[2] : 7  
array[3] : 10.7  
array[4] : -8
```

Saída

```
Somatorio: 19.20  
Produtorio: -13182.40
```

Exercício 4.4: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. Logo em seguida, o programa deve pedir o valor de um número inteiro. O programa deve apresentar ao usuário a mensagem “ACHEI” caso o valor seja encontrado em um determinado índice (posição) ou “NAO ACHEI” caso contrário.

Arquivo com a solução: [ex4.4.c](#)

Entrada

```
array[0] : 4  
array[1] : 5  
array[2] : 7  
array[3] : 10  
array[4] : -8  
Buscar por: 5
```

Saída

```
Indice 0: NAO ACHEI  
Indice 1: ACHEI  
Indice 2: NAO ACHEI  
Indice 3: NAO ACHEI  
Indice 4: NAO ACHEI
```

Exercício 4.5: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. Logo em seguida, o programa deve pedir o valor de um número inteiro. O programa deve contar quantas ocorrências do número fornecido foram encontradas no array, apresentando, ao final, essa contagem.

Arquivo com a solução: [ex4.5.c](#)

Entrada

```
array[0] : 4  
array[1] : 5  
array[2] : 7  
array[3] : 10  
array[4] : -8  
Buscar por: 5
```

Saída

0 array contem 1 ocorrencia do valor 5.

Entrada

```
array[0]: 7
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: 7
Buscar por: 7
```

Saída

0 array contem 3 ocorrencias do valor 7.

Entrada

```
array[0]: 4
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: -8
Buscar por: 0
```

Saída

0 array nao contem o valor 0.

Exercício 4.6: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. Logo em seguida, o programa deve pedir o valor de um número inteiro. O programa deve apresentar ao usuário todos os índices (posições) em que o valor fornecido foi encontrado no array.

Arquivo com a solução: [ex4.6.c](#)

Entrada

```
array[0]: 4  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: -8  
Buscar por: 5
```

Saída

O valor 5 foi encontrado no índice 1 do array.

Entrada

```
array[0]: 9  
array[1]: 5  
array[2]: 7  
array[3]: 9  
array[4]: 7  
Buscar por: 9
```

Saída

O valor 9 foi encontrado nos índices 0 e 3 do array.

Entrada

```
array[0]: 7  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: 7  
Buscar por: 7
```

Saída

O valor 7 foi encontrado nos índices 0, 2 e 4 do array.

Entrada

```
array[0]: 4
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: -8
Buscar por: 0
```

Saída

```
0 array nao contem o valor 0.
```

Exercício 4.7: Escreva um programa que preencha dois arrays de números inteiros de 5 posições com valores fornecidos pelo usuário. O programa deve preencher um terceiro array com a soma dos dois arrays preenchidos previamente e então exibir o array que contém a soma.

Arquivo com a solução: [ex4.7.c](#)

Entrada

```
Forneça os valores do primeiro array:
```

```
array1[0]: 5
array1[1]: 8
array1[2]: 7
array1[3]: 2
array1[4]: 8
```

```
Forneça os valores do segundo array:
```

```
array2[0]: 12
array2[1]: -10
array2[2]: -5
array2[3]: -20
array2[4]: 9
```

Saída

```
arraySoma[0] = 17  
arraySoma[1] = -2  
arraySoma[2] = 2  
arraySoma[3] = -18  
arraySoma[4] = 17
```

Exercício 4.8: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. O programa deve exibir os números pares desse array e depois os números ímpares, todos na ordem em que aparecem no array.

Arquivo com a solução: [ex4.8.c](#)

Entrada

```
array[0]: 4  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: -8
```

Saída

```
Numeros pares: 4 10 -8.  
Numeros impares: 5 7.
```

Entrada

```
array[0]: 4  
array[1]: 8  
array[2]: 6  
array[3]: 10  
array[4]: -8
```

Saída

```
Numeros pares: 4 8 6 10 -8.  
Numeros impares: nao ha.
```

Entrada

```
array[0] : 5  
array[1] : 9  
array[2] : 7  
array[3] : 13  
array[4] : -9
```

Saída

```
Numeros pares: nao ha.  
Numeros impares: 5 9 7 13 -9.
```

Exercício 4.9: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. O programa deve copiar os valores desse array para um segundo array, sendo que no segundo array, os valores serão inseridos de forma inversa. Ao final, o programa deve exibir os valores do array invertido.

Arquivo com a solução: [ex4.9.c](#)

Entrada

```
array[0] : 4  
array[1] : 8  
array[2] : 6  
array[3] : 10  
array[4] : -8
```

Saída

```
arrayInv[0] = -8  
arrayInv[1] = 10  
arrayInv[2] = 6  
arrayInv[3] = 8  
arrayInv[4] = 4
```

Exercício 4.10: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. Logo em seguida, o programa deve pedir o valor de um número inteiro. O programa deve copiar para um segundo array, todos os valores do primeiro array que são maiores que o último valor fornecido. Ao final, o programa deve exibir esses valores.

Arquivo com a solução: [ex4.10.c](#)**Entrada**

```
array[0]: 4  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: -8  
Copiar maiores que: 5
```

Saída

```
arrayCopia[0] = 7  
arrayCopia[1] = 10
```

Entrada

```
array[0]: 4  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: -8  
Copiar maiores que: 100
```

Saída

```
Nao houve copia!
```

Exercício 4.11: Escreva um programa para ler a quantidade de elementos que serão armazenados em um array de 10 posições de números inteiros. O programa deve aceitar apenas valores entre 1, inclusive, e 9, inclusive. Caso o seja fornecido um valor incorreto, ou seja, fora desse intervalo, o programa deve requisitar novamente a entrada da quantidade. Após a leitura de uma quantidade válida, ele deve ler a quantidade de elementos informados, armazenando-os no array a partir da primeira posição. Logo em seguida, o programa deve pedir o valor de um número inteiro. Esse número deve ser inserido na primeira posição do array. Antes da inserção, perceba que há a necessidade de deslocar os elementos existentes para a casa à direita. Por fim, o programa deve imprimir o array após o deslocamento e a inclusão.

Arquivo com a solução: [ex4.11.c](#)

Entrada

```
Quantidade de elementos (1 a 9): 20
Quantidade incorreta, forneça novamente!
Quantidade de elementos (1 a 9): 5
array[0]: 4
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: -8
Valor que sera inserido: 15
```

Saída

```
array[0] = 15
array[1] = 4
array[2] = 5
array[3] = 7
array[4] = 10
array[5] = -8
```

Exercício 4.12: Escreva um programa que preencha um array de números inteiros de 5 posições com valores fornecidos pelo usuário. O primeiro elemento do array deve ser “excluído”, deslocando para isso todos os elementos a partir da segunda posição para a esquerda. Por fim, o programa deve imprimir o array após a remoção.

Arquivo com a solução: [ex4.12.c](#)

Entrada

```
array[0]: 4
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: -8
```

Saída

```
array[0] = 5  
array[1] = 7  
array[2] = 10  
array[3] = -8
```

Exercício 4.13: Escreva um programa que preencha um array de números inteiros de 10 posições com valores fornecidos pelo usuário. Em seguida, o programa deve ler o índice de uma posição do array, ou seja, um valor de 0 a 9. Caso seja informado uma posição inválida, o programa deve informar o usuário e pedir novamente a posição. Após a leitura da posição válida, o programa deve “remover” do array o elemento contido na posição fornecida. Por fim, o programa deve imprimir o array após a remoção.

Arquivo com a solução: [ex4.13.c](#)

Entrada

```
array[0]: 4  
array[1]: 5  
array[2]: 7  
array[3]: 10  
array[4]: -8  
array[5]: 9  
array[6]: 10  
array[7]: 14  
array[8]: 3  
array[9]: 121  
Posicao a ser removida (0 a 9): 20  
Posicao invalida, forneça novamente!  
Posicao a ser removida (0 a 9): 5
```

Saída

```
array[0] = 4
array[1] = 5
array[2] = 7
array[3] = 10
array[4] = -8
array[5] = 10
array[6] = 14
array[7] = 3
array[8] = 121
```

Exercício 4.14: Escreva um programa que preencha um array de números inteiros de 10 posições com valores fornecidos pelo usuário. O programa deve remover do array todos os elementos que forem pares. Por fim, o programa deve imprimir o array após as remoções.

Arquivo com a solução: [ex4.14.c](#)

Entrada

```
array[0]: 4
array[1]: 5
array[2]: 7
array[3]: 10
array[4]: -8
array[5]: 9
array[6]: 10
array[7]: 14
array[8]: 3
array[9]: 121
```

Saída

```
array[0] = 5
array[1] = 7
array[2] = 9
array[3] = 3
array[4] = 121
```

Exercício 4.15: Escreva um programa que preencha dois arrays de números inteiros

de 5 posições com valores fornecidos pelo usuário. Um terceiro array deve ser preenchido, contendo a intersecção dos elementos contidos nos dois primeiros arrays, ou seja, os valores que são comuns aos dois. Nos dois arrays fornecidos, pode haver repetição de elementos, mas essa repetição não deve ser refletida no array de intersecção. Por fim, o programa deve imprimir o array que contém os valores comuns aos dois arrays fornecidos.

Arquivo com a solução: [ex4.15.c](#)

Entrada

Forneça os valores do primeiro array:

```
array1[0]: 5  
array1[1]: 8  
array1[2]: 7  
array1[3]: 2  
array1[4]: 8
```

Forneça os valores do segundo array:

```
array2[0]: 5  
array2[1]: -10  
array2[2]: -5  
array2[3]: 8  
array2[4]: 2
```

Saída

```
arrayInterseccao[0] = 5  
arrayInterseccao[1] = 8  
arrayInterseccao[2] = 2
```

Entrada

Forneça os valores do primeiro array:

```
array1[0]: 5  
array1[1]: 8  
array1[2]: 7  
array1[3]: 2  
array1[4]: 8
```

Forneça os valores do segundo array:

```
array2[0]: 12  
array2[1]: -10  
array2[2]: -5  
array2[3]: -20  
array2[4]: 9
```

Saída

Nao ha interseccao entre os elementos dos dois arrays
fornecidos!

4.3 Desafios

Desafio 4.1: Escreva um programa que preencha um array de números inteiros de 10 posições com valores fornecidos pelo usuário. O programa deve exibir os números pares desse array em ordem crescente e depois os números ímpares em ordem decrescente.

Arquivo com a solução: [de4.1.c](#)