

## Vectores

Vectores nada mais são que matrizes unidimensionais. Vectores são uma estrutura de dados muito utilizada. É importante notar que vectores, matrizes bidimensionais e matrizes de qualquer dimensão são caracterizadas por terem todos os elementos pertencentes ao mesmo tipo de dados. Para se declarar um vector podemos utilizar a seguinte forma geral:

```
tipo_da_variável nome_da_variável [tamanho];
```

Quando em C se escreve uma declaração como esta ele reserva um espaço na memória suficientemente grande para armazenar o número de células especificadas em tamanho. Por exemplo, se declararmos:

```
float exemplo [20];
```

O C irá reservar  $4 \times 20 = 80$  bytes (4 bytes para cada número). Estes bytes são reservados de maneira contígua. Na linguagem C a numeração começa sempre em zero. Isto significa que, no exemplo acima, os dados serão indexados de 0 a 19. Para ceder a cada um deles temos que escrever:

```
exemplo[0]  
exemplo[1]  
.  
.  
.  
exemplo[19]
```

Mas ninguém o impede de escrever:

```
exemplo[30]  
exemplo[103]
```

Porquê? Porque o C não verifica se o índice usado está dentro dos limites válidos. Este é um cuidado que tem que ter. Se o programador não tiver atenção com os limites de validade para os índices ele corre o risco de ter sobreescritas ou de ver o computador parar. Bugs terríveis podem surgir.

**Atente no seguinte exemplo:**

Este algoritmo é constituído por um vector de 5 posições e calcula a soma de quatro números introduzidos pelo utilizador e apresenta o resultado da soma.

```
#include<stdio.h>
main()
{
float vec[5]; /* Vector com 5 posições */
float soma=0;
int i;

for (i=1;i<=4;i++)
    {
    printf("digite um número ->");
    scanf("%f",&vec[i]);
    soma=soma+vec[i];
    }

printf ("a meida total , de: %5.0f",soma);

scanf("%d");
}
```

**Notas importantes:**

Para evitar erros, todos os dados directamente relacionados com o vector têm que ter o mesmo tipo de dados do vector.

**Por isso neste exemplo a soma é declarada como float.**

O vector tem sempre que ter pelo menos uma espaço a mais que a quantidade de números que vai absorver.

**Por isso neste exemplo o vector é declarado com 5 posições.**

**Explicação:**

```
float vec[5];
```

Declaração de um vector com 5 posições uma para um carácter nulo e as restantes para os números a serem introduzidos pelo utilizador.

Posição 1 ou i	Posição 2 ou i++	Posição 3 ou i+2	Posição 4 ou i+3	Posição 5 ou i+4

```
for (i=1;i<=4;i++)
    {
    printf("digite um número ->");
    scanf("%f",&vec[i]);
    soma=soma+vec[i];
    }
```

Nesta parte o programa pede ao utilizador por 4 vezes para introduzir um número, lê o número introduzido e atribui o valor introduzido à variável soma. Assim sendo se o utilizador introduzir os números 3 5 7 6 o nosso vector irá ter o seguinte aspecto:

Posição 1 ou i	Posição 2 ou i++	Posição 3 ou i+2	Posição 4 ou i+3	Posição 5 ou i+4
3	5	7	6	/0

**Exercício 1**

Elabore um algoritmo que calcule a soma de quatro notas introduzidas pelo utilizador e apresente a média das 4 notas. A média não deve ultrapassar as duas casas decimais.

```
#include<stdio.h>
main()
{
float vec[5];
float media, soma=0;
int i;

for (i=1;i<=4;i++)
    {
        printf("digite a %dª nota ->",i);
        scanf("%f",&vec[i]);
        soma=soma+vec[i];
    }
media=soma/4;
printf ("a meida total é de: %2.2f",media);

scanf("%d");
}
```

**Exercício 2**

Elabore um algoritmo que introduzidas 4 notas de um aluno calcule a média e indique qual o valor da terceira nota.

```
#include<stdio.h>
main()
{
float vec[5];
float media, soma=0;
int i;
for (i=1;i<=4;i++)
    {
        printf("digite a %dª nota ->",i);
        scanf("%f",&vec[i]);
        soma=soma+vec[i];
    }
media=soma/4;
printf ("a meida total , de: %2.2f",media);
printf ("\n a 3ª nota , -> %2.0f",vec[3]);
scanf("%d");
}
```

### Exercício 3

Altere o algoritmo anteriormente criado (**Exercício 2**) para que este pergunte ao utilizador quantas notas quer introduzir para o cálculo da média.

```
#include<stdio.h>
main()
{
float vec[10];
float media, soma=0;
int num, i;
printf("Quantas notas quer calcular");
scanf("%d",&num);

for (i=1;i<=num;i++)
{
printf("digite a %d| nota ->",i);
scanf("%f",&vec[i]);
soma=soma+vec[i];
}
media=soma/num;
printf ("a meida total , de: %2.2f",media);
printf ("\n a 3ª nota , -> %2.0f" ,vec[3]);
scanf("%d");
}
```

### Exercício 4

Até aqui os algoritmos mostram sempre a 3ª nota, vamos então alterar isso:

Altere o algoritmo anteriormente criado (**Exercício 3**) para que este pergunte ao utilizador qual a nota que quer mostrar.

```
#include<stdio.h>
main()
{
float vec[10];
float media, soma=0;
int num, most, i;
printf("Quantas notas quer calcular");
scanf("%d",&num);

printf("\n Qual a nota que quer mostrar");
scanf("%d",&most);

for (i=1;i<=num;i++)
{
printf("digite a %d| nota ->",i);
scanf("%f",&vec[i]);
soma=soma+vec[i];
}
media=soma/num;
printf ("a meida total , de: %2.2f",media);
printf ("\n a %d| nota , -> %2.0f" ,most,vec[most]);
scanf("%d");
}
```

Algoritmo que soma 5 aos números em posição ímpar do vector e multiplica por 5 os números em posição par no vector.

```
#include<stdio.h>
#include<conio.h>
main()
{
int veca[10], vecb[10];
int i;
clrscr();
for (i=1;i<=5;i++)
{
printf("digite um valor para o %dª ",i);
scanf("%d",&veca[i]);
}
for (i=1;i<=5;i++)
{
if(i%2==0)
vecb[i]=veca[i]*5;
else
vecb[i]=veca[i]+5;
}
for (i=1;i<=5;i++)
printf("veca[%d]=%d  vecb[%d]=%d\n",i,veca[i],i,vecb[i]);
scanf("%d");
}
```

- Neste algoritmo declaramos 2 vectores com 10 posições cada um, veca e vecb.

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9

- No primeiro for vamos pedir ao utilizador que digite números para preencher 5 "células" do nosso vector. Supondo que o utilizador digita os números em baixo indicados o nosso vector ficará:

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9
<b>3</b>	<b>7</b>	<b>12</b>	<b>16</b>	<b>21</b>	--	--	--	--	--

- No segundo for vamos multiplicar por 5 todos os números que ficarem numa posição par do nosso vector e vamos somar 5 a todos aqueles que ficarem numa posição ímpar. Assim:

1	2	3	4	5	6	7	8	9	10
i	I+1	I+2	I+3	I+4	I+5	I+6	I+7	I+8	I+9
<b>3</b>	<b>7</b>	<b>12</b>	<b>16</b>	<b>21</b>	--	--	--	--	--
<b>3+5=8</b>	<b>7*5=35</b>	<b>12+5=17</b>	<b>16*5=80</b>	<b>21+5=26</b>					

- No terceiro e último for fazemos o output do resultado.

**Exercício 5**

Altere o algoritmo anteriormente criado para que faça exactamente o mesmo mas desta vez deve utilizar como referencia os números do vector e não a posição dos números no vector.

```
#include<stdio.h>
#include<conio.h>
main()
{
int veca[10], vecb[10];
int i;
clrscr();
for (i=1;i<=5;i++)
    {
        printf("digite um valor para o %dª ",i);
        scanf("%d",&veca[i]);
    }
for (i=1;i<=5;i++)
    {
        if(veca[i]%2==0)
            vecb[i]=veca[i]*5;
        else
            vecb[i]=veca[i]+5;
    }
for (i=1;i<=5;i++)
    printf("veca[%d]=%d  vecb[%d]=%d\n",i,veca[i],i,vecb[i]);
scanf("%d");
}
```

**Exercício 6**

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e some todos os números em posição impar no vector.

```
#include<stdio.h>
main()
{
int a[5];
int soma,i;
soma=0;
for (i=1;i<=5;i++)
    {
        printf("digite o %dª valor",i);
        scanf("%d",&a[i]);
    }
for (i=1;i<=5;i++)
    {
        if (i%2!=0)
            soma=soma+a[i];
    }
printf("a soma , -> %d",soma);
scanf("%d");
}
```

**Exercício 7**

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e some todos os números impares do vector.

```
#include<stdio.h>
main()
{
int a[5];
int soma,i;
soma=0;
for (i=1;i<=5;i++)
{
printf("digite o %d$ valor",i);
scanf("%d",&a[i]);
}
for (i=1;i<=5;i++)
{
if (a[i]%2!=0)
soma=soma+a[i];
}
printf("a soma , -> %d",soma);
scanf("%d");
}
```

**Exercício 8**

Utilizando vectores elabore um algoritmo que peça ao utilizador para introduzir 5 números e apresente o resultado da soma dos números impares e pares do vector.

```
#include<stdio.h>
main()
{
int a[10];
int soma,soma2,i;
soma2=0;
soma=0;
for (i=1;i<=5;i++)
{
printf("digite o %d$ valor",i);
scanf("%d",&a[i]);
}
for (i=1;i<=5;i++)
{
if (a[i]%2!=0)
soma=soma+a[i];
else
soma2=soma2+a[i];
}
printf("a soma dos impares , -> %d",soma);
printf("\na soma dos pares , -> %d",soma2);

scanf("%d");
}
```

**Exemplo com vectores:**

```
#include <stdio.h>
main ()
{
int num[100]; /* Declara um vector de inteiros de 100 posições */
int count=0;
int totalnums;
do
    {
    printf ("\nDigite o numero (-999 p/ terminar): ");
    scanf ("%d",&num[count]);
    count++;
    } while (num[count-1]!=-999);
totalnums=count-1;
printf ("\n\n\n\t Os números que digitou foram:\n\n");
for (count=0;count<totalnums;count++)
    {
    printf (" %d",num[count]);
    }
}
```

No exemplo acima, o inteiro *count* é inicializado a 0. O programa pede pela entrada de números até que o utilizador digite a **Flag** -999. Os números são armazenados no vector **num**. A cada número armazenado, o contador do vector é incrementado para na próxima iteração escrever na próxima posição do vector. Quando o utilizador digita a flag, o programa abandona o primeiro loop e armazena o total de números gravados. Por fim, todos os números são impressos. É bom lembrar que nenhuma restrição é feita quanto a quantidade de números digitados. Se o utilizador digitar mais de 100 números, o programa tentará ler normalmente, mas o programa vai escreve-los numa parte não criada de memória, pois o espaço criado foi somente para 100 inteiros. Isto pode resultar nos mais variados erros, principalmente em tempo de execução..

**Exercício 9**

Utilize um vector 5 posições.

Reescreva o exemplo acima, realizando a cada leitura um teste para ver se a dimensão do vector não foi ultrapassada. Caso o utilizador atinja o limite máximo de números no vector o programa deve emitir uma mensagem para alertar o utilizador e obriga-lo a sair. Este programa deve ainda informar o utilizador quantos números ainda pode inserir à medida que este vai inserindo números.

```
#include <stdio.h>
#include <conio.h>
main ()
{
int num[5]; /* Declara um vector de inteiros de 100 posições */
int count=1;
int count2=0;
int totalnums;
printf("\tpara ver o resultado digite -999\n");

do

    {
        if(count<5)
printf (" Digite o %dº numero: ",count);
        else
printf("NAO PODE DIGITAR MAIS \n Digite -999 para ver o resultado ");

scanf ("%d",&num[count]);
count++;

        count2=5-count;
if (count2>0)
    {
printf(" Aten#Eo so pode digitar mais %d numeros\n ",count2);
    }

    }

while (num[count-1]!=-999);

totalnums=count-1;
printf ("\n\n\n\t Os numeros que digitou foram:\n\n");

for (count=1;count<totalnums;count++)
    {
printf (" %d",num[count]);
    }

scanf("%d");
}
```