



Algoritmos e Programação de Computadores

Vetores - Exercícios

Prof. Lucas Boaventura
lucas.boaventura@unb.br





Introdução

- LEMBRETE: Toda informação contida está armazenada na memória do computador

Endereço

Valor

...

...

0x7ffcc203070c	0
0x7ffcc2030708	8355
0x7ffcc2030704	4
0x7ffcc2030700	-1366160

...

...





Introdução

- LEMBRETE: As variáveis são mapeadas pelo compilador para determinada(s) região(ões)

Endereço Valor Variável

...

...

0x7ffcc203070c	0
0x7ffcc2030708	8355
0x7ffcc2030704	4
0x7ffcc2030700	-1366160

...

...

i
num





Introdução

- `num = 5;`

Endereço

...

Valor

...

Variável

0x7ffcc203070c	0
0x7ffcc2030708	8355
0x7ffcc2030704	5
0x7ffcc2030700	-1366160

...

...

i
num





Vetores: Soma

- Faça um programa que leia “N” números do teclado e realize a soma deles





Vetores: Soma

```
int main()
```

```
{
```

```
→ int soma;
```

```
int i;
```

```
int n;
```

```
scanf("%d", &n);
```

```
int nums[n];
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
    scanf("%d", &nums[i]);
```

```
}
```

```
soma = 0;
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
    soma += nums[i];
```

```
}
```

```
printf("%d\n", soma);
```

```
return 0;
```

```
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

LIXO

soma

0x7f..7f
8

LIXO

0x7f..7f
4

LIXO

0x7f..7
e0

LIXO

0x7f..7
e4

LIXO

0x7f..7
e8

LIXO

Saída:



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	LIXO	i
0x7f..7f 4	LIXO	
0x7f..7 e0	LIXO	
0x7f..7 e4	LIXO	
0x7f..7 e8	LIXO	



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	LIXO	i
0x7f..7f 4	LIXO	n
0x7f..7 e0	LIXO	
0x7f..7 e4	LIXO	
0x7f..7 e8	LIXO	



Vetores: Soma

```
int main()
```

```
{
```

```
    int soma;
```

```
    int i;
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    int nums[n];
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        scanf("%d", &nums[i]);
```

```
    }
```

```
    soma = 0;
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        soma += nums[i];
```

```
    }
```

```
    printf("%d\n", soma);
```

```
    return 0;
```

```
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

LIXO

soma

0x7f..7f
8

LIXO

i

0x7f..7f
4

3

n

0x7f..7
e0

LIXO

0x7f..7
e4

LIXO

0x7f..7
e8

LIXO

Saída:



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	LIXO	i
0x7f..7f 4	3	n
0x7f..7 e0	LIXO	nums[0]
0x7f..7 e4	LIXO	nums[1]
0x7f..7 e8	LIXO	nums[2]



Vetores: Soma

```
int main()  
{
```

```
    int soma;  
    int i;  
    int n;  
    scanf("%d", &n);  
    int nums[n];
```

→

```
    for (i = 0; i < n; i++)  
    {  
        scanf("%d", &nums[i]);  
    }
```

```
    soma = 0;  
    for (i = 0; i < n; i++)  
    {  
        soma += nums[i];  
    }  
    printf("%d\n", soma);  
    return 0;
```

```
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	0	i
0x7f..7f 4	3	n
0x7f..7 e0	LIXO	nums[0]
0x7f..7 e4	LIXO	nums[1]
0x7f..7 e8	LIXO	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	0	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	LIXO	nums[1]
0x7f..7 e8	LIXO	nums[2]



Vetores: Soma

```
int main()
```

```
{
```

```
    int soma;
```

```
    int i;
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    int nums[n];
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        scanf("%d", &nums[i]);
```

```
    }
```

```
    soma = 0;
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        soma += nums[i];
```

```
    }
```

```
    printf("%d\n", soma);
```

```
    return 0;
```

```
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

LIXO

soma

0x7f..7f
8

1

i

0x7f..7f
4

3

n

0x7f..7
e0

1

nums[
0]

0x7f..7
e4

LIXO

nums[
1]

0x7f..7
e8

LIXO

nums[
2]

Saída:



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	1	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	LIXO	nums[2]



Vetores: Soma

```
int main()
```

```
{
```

```
    int soma;
```

```
    int i;
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    int nums[n];
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        scanf("%d", &nums[i]);
```

```
    }
```

```
    soma = 0;
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        soma += nums[i];
```

```
    }
```

```
    printf("%d\n", soma);
```

```
    return 0;
```

```
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

LIXO

soma

0x7f..7f
8

2

i

0x7f..7f
4

3

n

0x7f..7
e0

1

nums[
0]

0x7f..7
e4

2

nums[
1]

0x7f..7
e8

LIXO

nums[
2]

Saída:



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	2	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
```

```
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];
```

→

```
    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }
```

```
    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
```

```
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	LIXO	soma
0x7f..7f 8	3	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	0	soma
0x7f..7f 8	3	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	0	soma
0x7f..7f 8	0	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	1	soma
0x7f..7f 8	0	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	1	soma
0x7f..7f 8	1	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de	Valor	Variável
Memória		
0x7f..7f c	3	soma
0x7f..7f 8	1	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	3	soma
0x7f..7f 8	2	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	6	soma
0x7f..7f 8	2	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	6	soma
0x7f..7f 8	3	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:
6

Endereço de	Valor	Variável
-------------	-------	----------

Memória

0x7f..7f c	6	soma
0x7f..7f 8	3	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

```
int main()
{
    int soma;
    int i;
    int n;
    scanf("%d", &n);
    int nums[n];

    for (i = 0; i < n; i++)
    {
        scanf("%d", &nums[i]);
    }

    soma = 0;
    for (i = 0; i < n; i++)
    {
        soma += nums[i];
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:
6

Endereço de Memória	Valor	Variável
0x7f..7f c	6	soma
0x7f..7f 8	3	i
0x7f..7f 4	3	n
0x7f..7 e0	1	nums[0]
0x7f..7 e4	2	nums[1]
0x7f..7 e8	3	nums[2]



Vetores: Soma

- No entanto, esse código está utilizando vetores de forma ineficiente!
- Será que para “ler N inteiros do teclado” nós precisamos ARMAZENAR N inteiros na memória?






Vetores: Soma Smart

```
int main()
```

```
{
```



```
int soma = 0;
```

```
int n;
```

```
int num;
```

```
scanf("%d", &n);
```

```
for (int i = 0; i < n; i++)
```

```
{
```

```
    scanf("%d", &num);
```

```
    soma += num;
```

```
}
```

```
printf("%d\n", soma);
```

```
return 0;
```

```
}
```

Endereço
de

Memória

Valor

Variável

0x7f..7f
c

LIXO

soma

0x7f..7f
8

LIXO

0x7f..7f
4

LIXO

0x7f..7
e0

LIXO

Saída:





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

0

soma

0x7f..7f
8

LIXO

n

0x7f..7f
4

LIXO

0x7f..7
e0

LIXO

Saída:





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

0

soma

0x7f..7f
8

LIXO

n

0x7f..7f
4

LIXO

num

0x7f..7
e0

LIXO

Saída:





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

0

soma

0x7f..7f
8

3

n

0x7f..7f
4

LIXO

num

0x7f..7
e0

LIXO

Saída:





Vetores: Soma Smart

```
int main()
```

```
{
```

```
    int soma = 0;
```

```
    int n;
```

```
    int num;
```

```
    scanf("%d", &n);
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        scanf("%d", &num);
```

```
        soma += num;
```

```
    }
```

```
    printf("%d\n", soma);
```

```
    return 0;
```

```
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

0

soma

0x7f..7f
8

3

n

0x7f..7f
4

LIXO

num

0x7f..7
e0

0

i

Saída:





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	0	soma
0x7f..7f 8	3	n
0x7f..7f 4	1	num
0x7f..7 e0	0	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	1	soma
0x7f..7f 8	3	n
0x7f..7f 4	1	num
0x7f..7 e0	0	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);
    → for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	1	soma
0x7f..7f 8	3	n
0x7f..7f 4	1	num
0x7f..7 e0	1	i

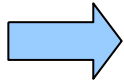




Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```



Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	1	soma
0x7f..7f 8	3	n
0x7f..7f 4	2	num
0x7f..7 e0	1	i

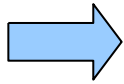




Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```



Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	3	soma
0x7f..7f 8	3	n
0x7f..7f 4	2	num
0x7f..7 e0	1	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Endereço
de

Valor

Variável

Memória

0x7f..7f
c

3

soma

0x7f..7f
8

3

n

0x7f..7f
4

2

num

0x7f..7
e0

2

i

Saída:

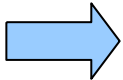




Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```



Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	3	soma
0x7f..7f 8	3	n
0x7f..7f 4	3	num
0x7f..7 e0	2	i

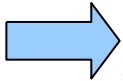




Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```



Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	6	soma
0x7f..7f 8	3	n
0x7f..7f 4	3	num
0x7f..7 e0	2	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    → for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:

Endereço de Memória	Valor	Variável
0x7f..7f c	6	soma
0x7f..7f 8	3	n
0x7f..7f 4	3	num
0x7f..7 e0	3	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:
6

Endereço de	Valor	Variável
----------------	-------	----------

Memória

0x7f..7f c	6	soma
0x7f..7f 8	3	n
0x7f..7f 4	3	num
0x7f..7 e0	3	i





Vetores: Soma Smart

```
int main()
{
    int soma = 0;
    int n;
    int num;
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &num);
        soma += num;
    }
    printf("%d\n", soma);
    return 0;
}
```

Saída:
6

Endereço de	Valor	Variável
-------------	-------	----------

Memória

0x7f..7f c	6	soma
0x7f..7f 8	3	n
0x7f..7f 4	3	num
0x7f..7 e0	3	i





Vetores

- No começo, veremos muitos exercícios de vetores
- Mais para frente, os alunos querem utilizar vetores sempre
- Nosso curso visa desenvolver Logica Computacional! É importante aprender a usar menos recursos e resolver o mesmo problema
- São necessários anos de treinamento, precisamos começar agora!





Dúvidas?

- lucas.boaventura@unb.br

