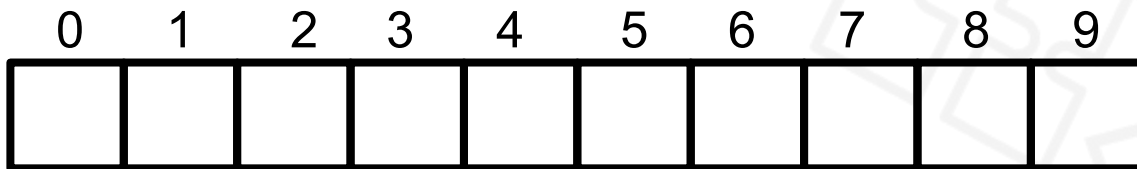


Arrays

Unidade I: Introdução às Collections

Definição

- Conjunto de variáveis que pode ser endereçada por um índice



- Posições válidas de 0 a $(n-1)$, logo, não podemos acessar posições negativas nem maiores que $(n-1)$

Declaração

- Acrescentamos um par de colchetes antes do nome da variável, por exemplo:

```
int[ ] array;  
double[ ] nota;
```

Definição do Tamanho

- Utilizamos o operador **new** conforme mostrado nos exemplos abaixo:

```
ind = new int[10];  
nota = new double[70];
```

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];

vet[0] = 5;
vet[1] = 9;
vet[2] = 312;

vet[0] = vet[1] + vet[2];
vet[1]--;

int pos = 0;
vet[pos] = 3;

for (int i = 0; i < 3; i++){
    vet[i] = int.Parse(Console.ReadLine());
}

for (int i = 0; i < 3; i++){
    Console.WriteLine(vet[i]);
}
```

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;  
vet[1] = 9;  
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];  
vet[1]--;
```

```
int pos = 0;  
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

Quantas posições nosso
arrays possui?

Quais são as posições
disponíveis?

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;  
vet[1] = 9;  
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];  
vet[1]--;
```

```
int pos = 0;  
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	0
1	0
2	0

Quantas posições nosso
arrays possui?

Resposta: 3

Quais são as posições
disponíveis?

Resposta: 0, 1 e 2

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;  
vet[1] = 9;  
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];  
vet[1]--;
```

```
int pos = 0;  
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

O que acontece se
acessarmos uma
posição negativa ou
maior que 3?

	vet
0	0
1	0
2	0

Acesso ao Conteúdo de uma Posição

	vet
0	0
1	0
2	0

```
int[] vet = new int [3];
```

```
vet[0] = 5;  
vet[1] = 9;  
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];  
vet[1]--;
```

```
int pos = 0;  
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

O que acontece se
acessarmos uma
posição negativa ou
maior que 3?

**Resposta: Temos uma
exceção**

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	0
2	0

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	5
1	9
2	0

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	5
1	9
2	312

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	321
1	9
2	312

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	321
1	8
2	312

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	3
1	8
2	312

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

Atenção: *pos* é
diferente de `vet[pos]`

	vet
0	3
1	8
2	312

Acesso ao Conteúdo de uma Posição

	vet
0	3
1	8
2	312

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
0

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

true or false?

	vet
0	3
1	8
2	312

i
0

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	8
2	312

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
0

Usuário digita 5

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	8
2	312

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++) {
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
1

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

true or false?

	vet
0	5
1	8
2	312

i
1

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

	vet
0	5
1	10
2	312

i
1

Usuário digita 10

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	312

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++) {
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
2

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

true or false?

	vet
0	5
1	10
2	312

i
2

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
2

Usuário digita 15

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++) {
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
3

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

true or false?

	vet
0	5
1	10
2	15

i
3

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

i
0

TELA

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	5
1	10
2	15

true or false?

i
0

TELA

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	5
1	10
2	15

i
0

TELA
5

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

i
1

TELA
5

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

true or false?

	vet
0	5
1	10
2	15

i
1

TELA
5

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];

vet[0] = 5;
vet[1] = 9;
vet[2] = 312;

vet[0] = vet[1] + vet[2];
vet[1]--;

int pos = 0;
vet[pos] = 3;

for (int i = 0; i < 3; i++){
    vet[i] = int.Parse(Console.ReadLine());
}

for (int i = 0; i < 3; i++){
    Console.WriteLine(vet[i]);
}
```

i
1

TELA
5 10

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

	vet
0	5
1	10
2	15

i
2

TELA
5 10

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

true or false?

	vet
0	5
1	10
2	15

i
2

TELA
5 10

Acesso ao Conteúdo de uma Posição

	vet
0	5
1	10
2	15

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){
```

```
    vet[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
for (int i = 0; i < 3; i++){
```

```
    Console.WriteLine(vet[i]);
```

```
}
```

i
2

TELA
5
10
15

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

	vet
0	5
1	10
2	15

i
3

TELA
5
10
15

Acesso ao Conteúdo de uma Posição

```
int[] vet = new int [3];
```

```
vet[0] = 5;
```

```
vet[1] = 9;
```

```
vet[2] = 312;
```

```
vet[0] = vet[1] + vet[2];
```

```
vet[1]--;
```

```
int pos = 0;
```

```
vet[pos] = 3;
```

```
for (int i = 0; i < 3; i++){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

```
for (int i = 0; i < 3; i++){  
    Console.WriteLine(vet[i]);  
}
```

true or false?

i

3

TELA

5

10

15

	vet
0	5
1	10
2	15

Index Out of Range Exception

- Acesso a posições maiores que $(n-1)$

```
int n = 10;  
int[] vet = new int[n];  
...  
for (int i = n; i >= 0; i--){  
    vet[i] = int.Parse(Console.ReadLine());  
}
```

Index Out of Range Exception

- Acesso a posições negativas

```
int n = 10;  
int[] vet = new int[n];  
  
...  
for (int i = n-1; i >= 0; i--){  
    vet[i-1] = int.Parse(Console.ReadLine());  
}
```


Exercício

- Faça um programa que leia a nota e o nome de n alunos e mostre na tela o nome daqueles que ficaram acima da média do grupo, sendo n um número inteiro lido do teclado