

Algoritmos e Estruturas de Dados I

Prof. Lúcio Mauro Pereira

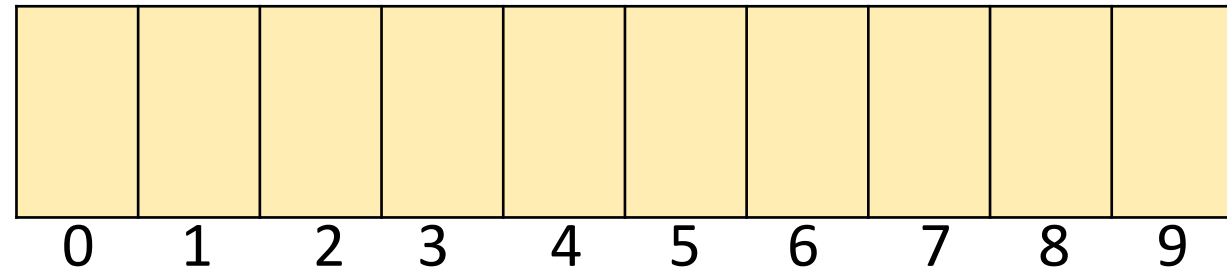
10 e 11/10/2024

Aulas anteriores

Arranjo expressando uma coleção

Vetor como coleção de dados

```
int main()
{
    int vetor[] = { 10,20,30,40,50,60,70,80,90,100 };
    escrevaArray( vetor );
    return 0;
}
// fim main()
```



Vetor como coleção de dados

```
int main()
{
    int vetor[] = { 10,20,30,40,50,60,70,80,90,100 };
    escrevaArray( vetor );
    return 0;
}
// fim main()
```

vetor

10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9

Vetor como coleção de dados

```
void escrevaArray(float A[], int TAM)
{
    for( int i=0 ; i < TAM ; i++ )
    {
        printf("\nA[%i] = %f", i+1, A[i]);
    }
}
```

```
int main()
{
    int vetor[] = { 10,20,30,40,50,60,70,80,90,100 };
    escrevaArray( vetor, 10 );
    return 0;
}
// fim main()
```

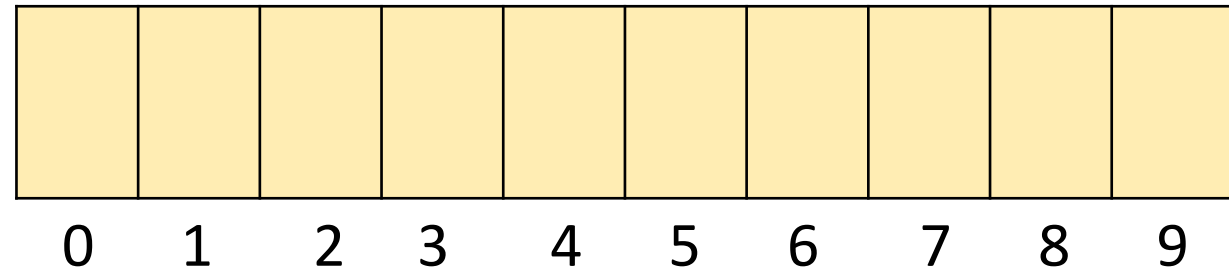
vetor

10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9

Arranjo com valores lidos

Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

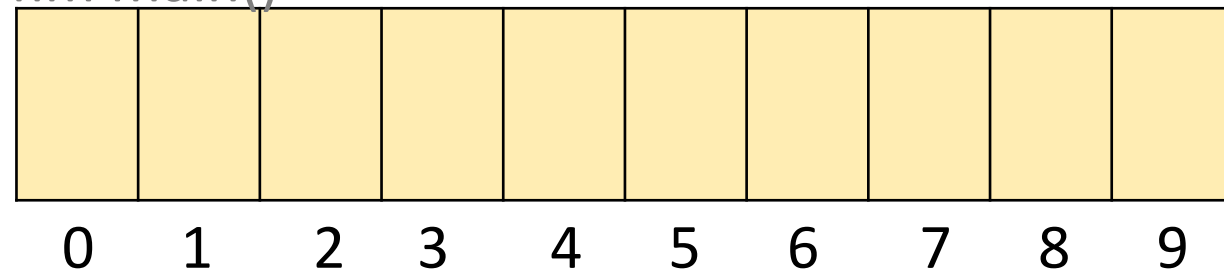
```
void leialdades(int idades[])
{
    for( int i=0 ; i<=9 ; i++ )
    {
        printf("\nDigite a %i.a idade: ", i+1 );
        scanf ( "%i", &idades[i] );
    }
}
```



Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

```
void leialdades(int idades[])
{
    for( int i=0 ; i<=9 ; i++ )
    {
        printf("\nDigite a %i.a idade: ", i+1 );
        scanf ( "%i", &idades[i] );
    }
}
```

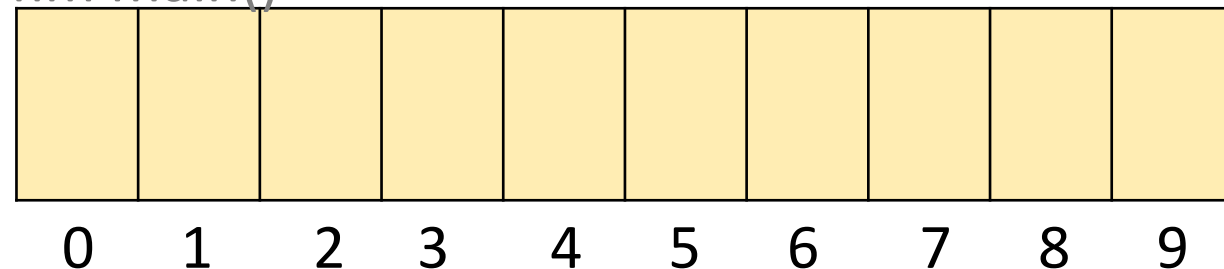
```
int main()
{
    int idades[10];
    leialdades( idades );
    float media = medialdades( idades );
    printf("\nMedia = %f", media );
    print("\n%i acima", acimaMedia(idades,media) );
    return 0;
}
// fim main()
```



Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

```
void leialdades(int idades[])  
{  
    for( int i=0 ; i<10 ; i++ )  
    {  
        printf("\nDigite a %i.a idade: ", i+1 );  
        scanf ( "%i", &idades[i] );  
    }  
}
```

```
int main()  
{  
    int idades[10];  
    leialdades( idades );  
    float media = medialdades( idades );  
    printf("\nMedia = %f", media );  
    print("\n%i acima", acimaMedia(idades,media) );  
    return 0;  
}  
} // fim main()
```



Tamanho definido em constante global

Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

```
void leialdades(int idades[])
{
    for( int i=0 ; i<      ; i++ )
    {
        printf("\nDigite a %i.a idade: ", i+1 );
        scanf ( "%i", &idades[i] );
    }
}
```

```
const int MAX = 10;
```

```
int main()
{
    int idades[    ];
    leialdades( idades );
    float media = medialdades( idades );
    printf("\nMedia = %f", media );
    print("\n%i acima", acimaMedia(idades,media) );
    return 0;
}
} // fim main()
```

Tamanho definido em #define

Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

```
void leialdades(int idades[])
{
    for( int i=0 ; i<      ; i++ )
    {
        printf("\nDigite a %i.a idade: ", i+1 );
        scanf ( "%i", &idades[i] );
    }
}
```

```
#define MAX 10
```

```
int main()
{
    int idades[    ];
    leialdades( idades );
    float media = medialdades( idades );
    printf("\nMedia = %f", media );
    print("\n%i acima", acimaMedia(idades,media) );
    return 0;
}
} // fim main()
```


Tamanho definido em valor lido

Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

```
void leialdades(int idades[])
{
    for( int i=0 ; i<      ; i++ )
    {
        printf("\nDigite a %i.a idade: ", i+1 );
        scanf ( "%i", &idades[i] );
    }
}
```

```
int MAX ;

int main()
{
    MAX =
    int idades[      ];
    leialdades( idades );
    float media = medialdades( idades );
    printf("\nMedia = %f", media );
    print("\n%i acima", acimaMedia(idades,media) );
    return 0;
}
} // fim main()
```

Tamanho estático, interrupção com *flag*

Ler idade da turma. *Flag*: idade igual a 0.
Identificar o número de idades acima da média.

```
void leiaIdades(int idades[])  
{  
  
    {  
  
        printf("\nDigite a %i.a idade: ", i+1 );  
        scanf ( "%i", &idades[i] );  
    }  
}
```

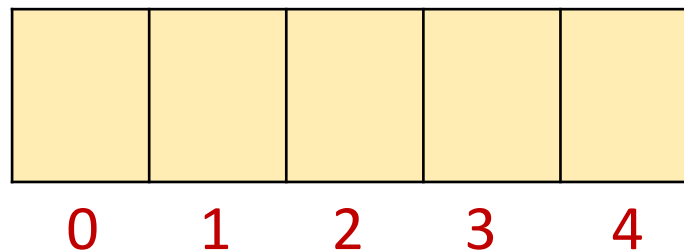
Desafios extra-classe

1) Função que identifica a maior idade

```
float maiorValor(float vetor[], int tamanho)
{
    return ;
}
```

```
int main()
{
    int arranjo[] = { };
    printf("\nMaior valor = %i", maiorValor(arranjo,5));
    return 0;
}
// fim main()
```

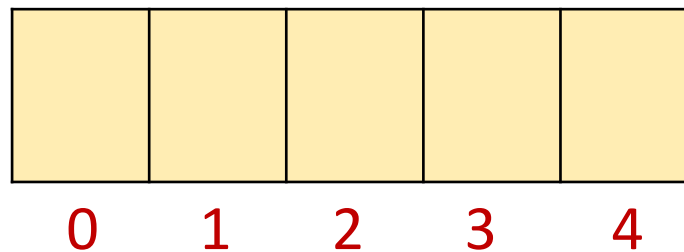
arranjo



2) Função que identifique a amplitude das idades

```
int main()
{
    int arranjo[] = {                };
    printf("\nAmplitude = %i", amplitude(arranjo,5));
    return 0;
}
// fim main()
```

amplitude



3) Função que identifica o percentual de maiores de idade

```
float maiorValor(float vetor[], int tamanho)
{

    return ;
}
```