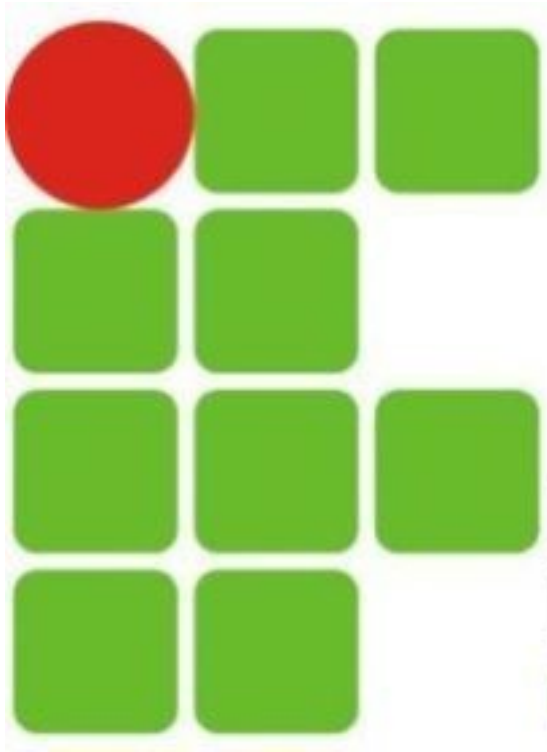


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA

LINGUAGEM DE PROGRAMAÇÃO

ARRAYS EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- As variáveis declaradas até agora são capazes de armazenar um único valor por vez.
- Assim, para armazenar mais de um valor, é preciso usar mais de uma variável.
- Imagine o seguinte problema: leia as notas de uma turma de cinco estudantes e depois imprima as notas que são maiores do que a média da turma.
- A solução apresentada é inviável para uma turma de 100 alunos.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
  float n1,n2,n3,n4,n5;
  printf("Digite a nota de 5 estudantes: ");
  scanf("%f",&n1);
  scanf("%f",&n2);
  scanf("%f",&n3);
  scanf("%f",&n4);
  scanf("%f",&n5);
  float media = (n1+n2+n3+n4+n5;)/5.0;
  if(n1 > media) printf("nota: %f\n",n1);
  if(n2 > media) printf("nota: %f\n",n2);
  if(n3 > media) printf("nota: %f\n",n3);
  if(n4 > media) printf("nota: %f\n",n4);
  if(n5 > media) printf("nota: %f\n",n5);
  system("pause");
  return 0;
}
```

ARRAYS EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Um array ou "vetor" é a forma mais simples e comum de dados estruturados da linguagem C.
- Trata-se simplesmente de um conjunto de variáveis de um mesmo tipo, com a vantagem de estarem todas associadas ao mesmo nome e igualmente acessíveis por um índice.
- Em linguagem C, a declaração de um array segue esta forma geral:

```
tipo_dado nome_array[tamanho];
```

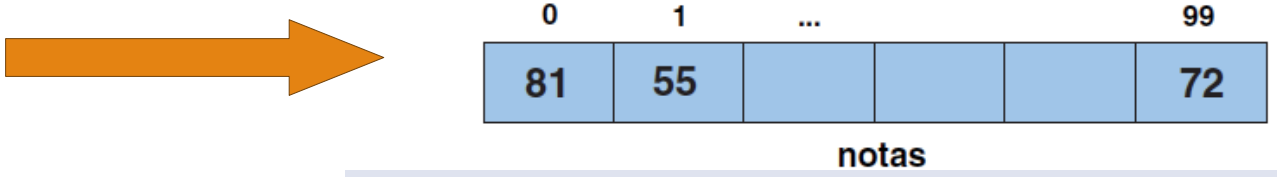
```
#include <stdio.h>
#include <stdlib.h>
#define N 100
int main(){
  int n = 5;
  float F[N+1];//Correto: expressão inteira e constante
  char texto[30];//Correto: valor inteiro
  int Vetor[n];//Errado: n é variável
  int V[4.5];//Errado: 4.5 não é inteiro
  system("pause");
  return 0;
}
```

ARRAYS EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- O acesso ao valor de cada nota é feito utilizando um índice

```
float notas[100];
notas[0] = 81;
notas[1] = 55;
...
notas[99] = 72;
```




```
#include <stdio.h>
#include <stdlib.h>
int main(){
  int notas[100];
  int i;
  for (i = 0; i < 100; i++){
     printf("Digite a nota do aluno %d",i);
     scanf("%d",&notas[i]);
  }
  system("pause");
  return 0;
}
```

ARRAYS EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- cada posição do array possui todas as características de uma variável.
- Isso significa que ela pode aparecer em comandos de entrada e saída de dados

```
scanf("%d",&notas[5]);//comando de leitura
notas[0] = 10;//comando de atribuição
notas[1] = notas[5] + notas[0];//expressão
```

- Na linguagem C, a numeração do índice do array começa sempre do ZERO e termina sempre em N-1, em que N é o número de elementos definido na declaração do array.

STRING EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- String é o nome que usamos para definir uma sequência de caracteres adjacentes na memória do computador.
- Essa sequência de caracteres, que pode ser uma palavra ou frase, é armazenada na memória do computador na forma de um array do tipo char.
- Por ser a string um array de caracteres, sua declaração segue as mesmas regras da declaração de um array convencional:

```
char str[6];
```

- Para sua inicialização, pode-se usar o mesmo princípio definido na inicialização de vetores

```
char str [10] = { 'J', 'o', 'a', 'o', '\0' };
```

- A inicialização de strings também pode ser feita por meio de aspas duplas:

```
char str [10] = "Joao";
```

TRABALHANDO COM STRING EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- O primeiro cuidado que temos que tomar ao trabalhar com strings é na operação de atribuição.
- Strings são arrays. Portanto, não se pode fazer atribuição de strings.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
  char str1[20] = "Hello World";
  char str2[20];

  str1 = str2; //ERRADO!

  system("pause");
  return 0;
}
```

- C não suporta a atribuição de um array para outro.
- Para atribuir o conteúdo de uma string a outra,deve-se copiar a string, elemento por elemento, para a outra string

```
#include <stdio.h>
#include <stdlib.h>
int main(){
  int i;
  char str1[20] = "Hello World";
  char str2[20];
  for (i = 0; str1[i]!='\0'; i++)
     str2[i] = str1[i];
  str2[i] = '\0';
  system("pause");
  return 0;
}
```

TRABALHANDO COM STRING EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Infelizmente, esse tipo de manipulação de arrays não é muito prático quando estamos trabalhando com palavras.
- Felizmente, a biblioteca-padrão da linguagem C possui funções especialmente desenvolvidas para a manipulação de strings na biblioteca <string.h>.
- gets(), faz a leitura do teclado considerando todos os caracteres digitados (incluindo os espaços) até encontrar uma tecla enter:

```
char str[20];

gets(str);
```

- Printf, faz a impressão da string na tela. Usando formato de dados %s

```
char str[20] = "Hello World";

printf("%s",str);
```

TRABALHANDO COM STRING EM C

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA BAHIA

- Atividade para a semana: busquem informações sobre as funções:
  - fgets(str, tamanho, stdin) → ler uma string definindo a quantidade de caracteres
  - setbuf(stdin, null) → limpar buffer do teclado antes de uma leitura
  - fputs(str,stdout) → apresenta uma string na tela
  - strlen(str) → apresenta o tamanho de uma string
  - strcpy(destino,origem) → compia uma string para outra
  - strcat(destino, origem) → concatena duas strings e armazena o resultado na primeira
  - strcmp(str1,str2) → compara duas strings. (0 se forem iguais, 1 se forem diferentes)