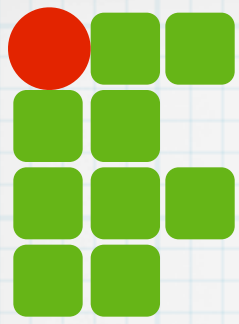


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Algoritmos

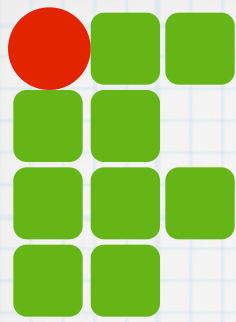
ANSI C - Arrays

Copyright © 2014 IFRN



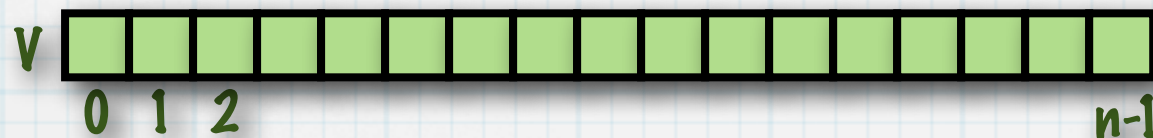
Agenda

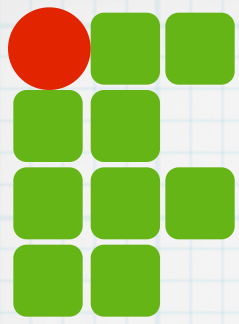
- * Arrays
- * Arrays como parâmetros
- * Exercícios



Arrays

- * Um array é um agregado homogêneo de elementos
 - * Única variável, referenciando uma área contínua de memória
 - * Um índice indica posição de um elemento no array
 - * índice representa deslocamento a partir do início da área de memória
 - * Em C, primeiro índice é o zero



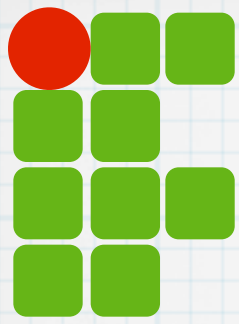


Arrays

* Declaração

- * Na declaração de um array é necessário informar a quantidade de elementos que o compõe e o tipo dos elementos
- * Array com 10 elementos inteiros

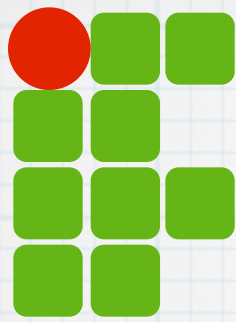
```
int x[10];
```

Arrays

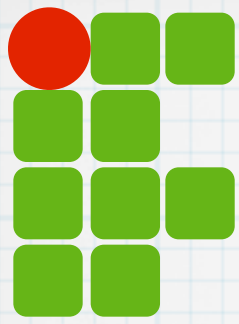
- * Acesso aos elementos através do índice
 - * Inteiro entre 0 e n-1
 - * O operador de indexação `[]` é utilizado para referenciar cada elemento

```
int x[10];  
...  
x[0] = 10;  
x[1] = 20;  
x[2] = x[0] + x[1];
```



Arrays

```
#include <stdio.h>
int main (int argc, char **argv){
    int i, x[10];
    for (i = 0 ; i < 10 ; i++){
        scanf("%d",&x[i]);
    }
    for (i = 0 ; i < 10 ; i++){
        printf("%d ",x[i]);
    }
    return 0;
}
```

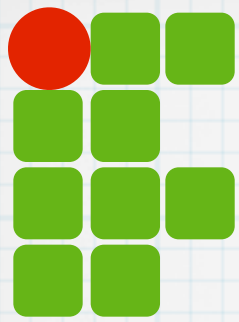


índices de arrays

- * Um índice pode ser uma expressão
- * Pode conter variáveis
- * Deve ser um valor inteiro

```
x[i]=x[i-1]*x[i+1];
```

```
x[i+y*k]=x[i/k]*3;
```

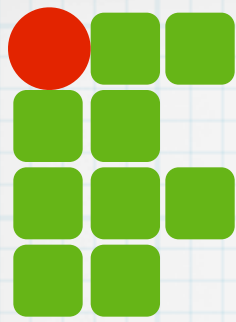


índices de arrays

* Fibonacci com arrays

* OBS: Uso de memória

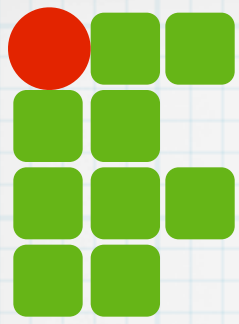
```
#include <stdio.h>
int main (int argc, char **argv){
    int i, x[1000];
    x[0]=1;
    x[1]=1;
    for (i = 2 ; i < 1000 ; i++){
        x[i]=x[i-1]+x[i-2];
    }
    for (i = 0 ; i < 1000 ; i++){
        printf("%d ",x[i]);
    }
    return 0;
}
```

Exemplo

* Maior elemento de um array

```
#include <stdio.h>
int main (int argc, char **argv){
    int i,maior,indiceMaior, x[100];
    for (i = 0 ; i < 100 ; i++){
        scanf("%d",&x[i]);
    }
    indiceMaior=0;
    maior=x[0];
    for (i = 1 ; i < 100 ; i++){
        if(x[i]>maior){
            indiceMaior=i;
            maior=x[i];
        }
    }
    printf("O maior elemento é %d e está no índice %d\n",maior,indiceMaior);
    return 0;
}
```



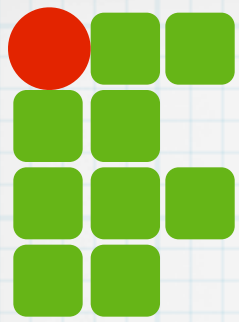
Parâmetros

* Valor

- * Como vimos, o valor é passado e a função não altera o valor da variável "original"

* Referência

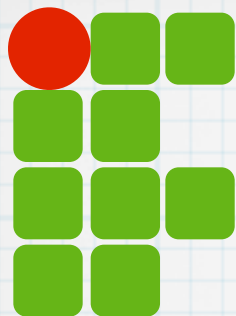
- * Passamos um referência a uma variável
- * Alterações modificam o valor da variável passada
- * Veremos como fazer isso usando ponteiros



Arrays como parâmetros

- * Podemos passar arrays como parâmetros
- * **CUIDADO:** alterar valores dos elementos de um array altera o valor original

```
void duplicaValores(int a[10]) {  
    int i;  
    for (i = 0; i < 10; i++) {  
        a[i] = a[i] * 2;  
    }  
}  
  
int main(int argc, char **argv) {  
    int a1[10];  
    int i;  
    for (i = 0; i < 10; i++) {  
        a1[i] = i;  
    }  
    duplicaValores(a1);  
    for (i = 0; i < 10; i++) {  
        printf("%d ", a1[i]);  
    }  
    return 0;  
}
```



Dúvidas?