



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA



Instituto Federal da Bahia

Análise e Desenvolvimento de Sistemas

INF029 – Laboratório de Programação

Aula Inaugural

Prof. Dr. Renato L. Novais
renato@ifba.edu.br

Agenda



- Revisão
- Debug

- Programa simples para trocar duas variáveis
- Programa para calcular a serie de Fibonacci
 - cada termo subsequente (numero de Fibonacci) corresponde a soma dos dois anteriores
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...
(podendo ser omitido o zero inicial).

$$F_n = F_{n-1} + F_{n-2},$$

$$F_1 = 1, F_2 = 1.$$

Revisão C



- Faça um programa no qual o usuário vai informar 10 números. Você deve verificar qual deles é primo. Depois da verificação de todos eles você deve imprimir:
 - Posição – número – sim/não
- Faça um programa no qual o usuário vai informar duas os valores de duas matrizes 3x2 e 2x3. Você deve apresentar uma terceira matriz que é o resultado da multiplicação das duas matrizes iniciais.

1	2
3	4
5	6

 x

0	1	3
2	1	0

 =

4	3	3
8	7	9
12	11	15

- Variáveis globais: acessíveis de qualquer arquivo
 - `int count; //fora de qualquer funcao`
- Static globais: acessível só no próprio arquivo
 - `static int count`
- Static locais: funciona como global. Existe durante toda a execução do programa, mas é acessível apenas dentro do bloco/função que a define

Struct em C



- Criar uma estrutura x com três campos

```
struct {  
    int dia;  
    int mes;  
    int ano;  
} x;
```

- Dando um nome a estrutura

```
struct dma {  
    int dia;  
    int mes;  
    int ano;  
};  
struct dma x; /* um registro x do tipo dma */  
struct dma y; /* um registro y do tipo dma */
```

- Atribuindo valores

```
x.dia = 31;  
x.mes = 8;  
x.ano = 1998;
```

Fonte: <http://www.ime.usp.br/~pf/algoritmos/aulas/stru.html>

Exercício Struct



- Crie um struct para definir uma pessoa com nome, id, sexo
- Crie um vetor de 10 posições, que seja do tipo pessoa
- Cadastre cada uma das pessoas
- Imprima cada uma das pessoas

Exercício

```
#include <stdio.h>
#define size 2

struct s_pessoa{
    int id;
    char nome [21];
    char sexo;
    int idade;
};

struct s_pessoa pessoas[size];

void trabalharComStruct(){
    //int j = 0;
    //char nome[20];
    char temp;
    for (int i = 0; i < size; ++i){
        pessoas[i].id = i+1;
        printf("Digite o nome da Pessoa (%d): ",pessoas[i].id);
        fflush(stdin);
        fgets(pessoas[i].nome, 21, stdin);
        strtok(pessoas[i].nome, "\n"); // Remove o \n que o fgets captura

        printf("Digite o sexo: ");
        scanf("%c",&pessoas[i].sexo);

        printf("Digite a idade: ");
        scanf("%d%c",&pessoas[i].idade);
    }

    printf("\nCadastrados");
    for (int i = 0; i < size; i++){
        printf("\n-----");
        printf("\nId: %d - Nome: %s - Sexo: %c - Idade: %d\n",pessoas[i].id,
            pessoas[i].nome,pessoas[i].sexo, pessoas[i].idade);
    }
}
```


Ordenação: Selection



- Busca o i-menor elemento e coloca na posição i

10	8	6	2	16	4	18	11	14	12
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2	8	6	10	16	4	18	11	14	12
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2	4	6	10	16	8	18	11	14	12
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2	4	6	10	16	8	18	11	14	12
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

O processo se repete até o fim da ordenação

Exemplo de ordenação Selection Sort
fonte da imagem: bdfatec.blogspot.com.br

Selection em C



```
void selection_sort(int num[], int tam)
{
    int i, j, min, aux;
    for (i = 0; i < (tam-1); i++)
    {
        min = i;
        for (j = (i+1); j < tam; j++) {
            if(num[j] < num[min]) {
                min = j;
            }
        }
        if (i != min) {
            aux = num[i];
            num[i] = num[min];
            num[min] = aux;
        }
    }
}
```

Selection em C para String



```
void ordenar_selecao_nome() {
    int i,j,n;
    for(i=0; i<n-1; i++) {
        for(j=i+1; j<n; j++) {
            if(strcmp(vetor[i], vetor[j])>0) {
                strcpy(aux_char, vetor[i]);
                strcpy(vetor[i], vetor[j]);
                strcpy(vetor[j], aux_char);
            }
        }
    }
}
```

Debug



- On the fly....

Exercício



- Fazer um programa que gerencie o cadastro de pessoas (id, nome, sexo, idade). O programa deve ser capaz de cadastrar até 10 pessoas
- O sistema deve gerar identificador automaticamente
- O programa deve ter as seguintes opções
 - 1. Cadastrar
 - 2. Excluir (usuário informa o nome)
 - 3. Listar
 - 4. Listar ordenado por: a) Nome b) idade
 - 5. Mostrar maiores de idade
 - 0. Sair