

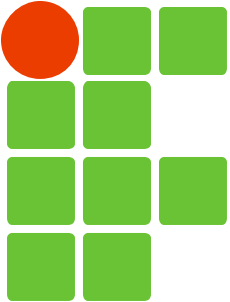
INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Apresentação da Disciplina

Estruturas de Dados

Prof. Victorio Albani de Carvalho



**INSTITUTO FEDERAL
ESPIRITO SANTO**

CAMPUS COLATINA

Conteúdo

Conceito de Tipos Abstratos de Dados

Pilhas

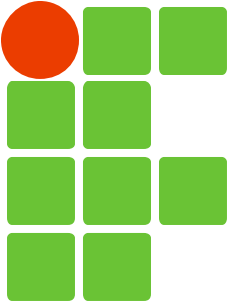
Filas

Listas

Árvores Binárias

Árvores AVL

Introdução a Grafos

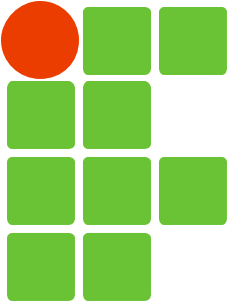


**INSTITUTO FEDERAL
ESPIRITO SANTO**

CAMPUS COLATINA

Avaliações

- 4 a 5 Trabalhos: 70 pontos;
- 1 Prova: 30 pontos;

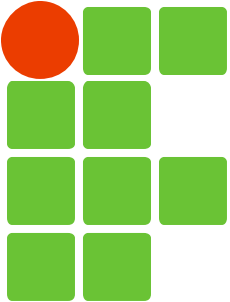


**INSTITUTO FEDERAL
ESPIRITO SANTO**

CAMPUS COLATINA

Estratégias

- Materiais disponíveis no Moodle;
- Aulas presenciais.

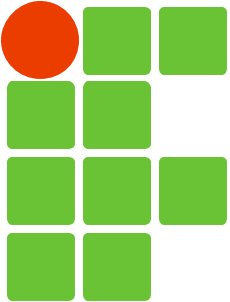


INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Conhecimentos Necessários

- Implementação em C
 - Conceitos básicos como comandos de decisão e repetição
 - Vetores e Matrizes
 - Structs
 - Funções (passagem de parâmetros)
 - **Alocação dinâmica de memória**



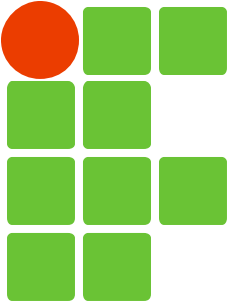
INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Avaliação Diagnóstica

- Objetivo: Identificar o nível de conhecimento da turma em programação
- Questionários no Moodle!

Responder individualmente agora...



INSTITUTO FEDERAL
ESPIRITO SANTO

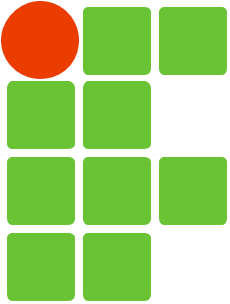
CAMPUS COLATINA

Diagnóstico

```
int main(){  
    int v[5],i;  
    for (i=0; i<=5;i++)  
        v[i]=i;  
}
```

Sobre o código acima, escrito em linguagem C, pode-se afirmar que:

- a) O código apresentará erro de compilação, pois acessa o índice 0 (zero) do vetor que é inválido. A indexação de vetores em C começa em 1 (um).
- b) O código está manipulando corretamente o vetor, pois o vetor declarado tem 5 posições logo os índices vão de 0(zero) a 5(cinco).
- c) O código pode apresentar erro durante a execução pois acessa o índice 5(cinco) do vetor, sendo que o maior índice válido seria o índice 4(quatro).
- d) O código apresenta erro pois tenta atribuir um valor inteiro a um vetor.



INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Diagnóstico

```
int main(){
    int m[3][3];
    int i, j, k=1;
    for (i=0; i<3;i++)
        for(j=0;j<3;j++){
            m[i][j]=k;
            k++;
        }
    for (i=0; i<3;i++)
        for(j=0;j<3;j++)
            if (i==j)
                printf("%d; ", mat[i][j]);

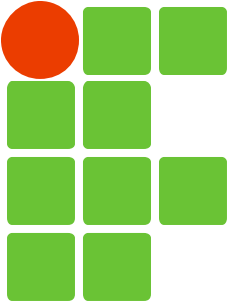
    return 0;
}
```

O que será impresso?

gnóstico

```
#include<stdio.h>
void f1(int n1,int n2)
{
    int temp;
    temp = n1;
    n1 = n2;
    n2 = temp;
}
void f2(int *n1,int *n2)
{
    int temp;
    temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}
int main() {
    int num1=50,num2=70;
    f1(num1,num2);
    printf("%d %d",num1, num2);
    num1=50;
    num2=70;
    f2(&num1,&num2);
    printf(" %d %d\n",num1, num2);
    return(0);
}
```

O que será impresso?



**INSTITUTO FEDERAL
ESPIRITO SANTO**

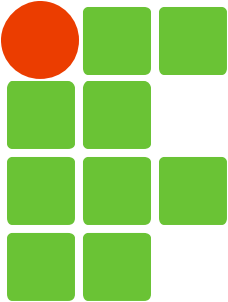
CAMPUS COLATINA

Diagnóstico

```
int dobro(int x){  
    return 2*x;  
}
```

```
int main(){  
    int a = 5;  
    dobro(a);  
    printf("O valor final de a eh %d", a);  
    return 0;  
}
```

Qual valor será impresso?



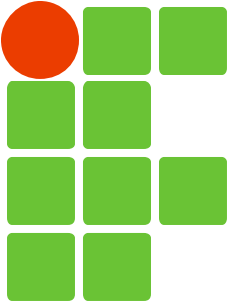
INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Diagnóstico

```
int f(int x)
{
    if (x==1)
        return x;
    else
        return x+f(x-1);
}
```

- a) Caso a função seja chamada passando o valor 4 (ou seja, $f(4)$), o retorno da execução será o valor 7.
- b) Caso a função seja chamada passando o valor 0 (ou seja, $f(0)$), a execução nunca terminará, ou seja teremos uma recursão infinita.
- c) Independente do valor passado como parâmetro a função nunca dará retorno.
- d) Caso a função seja chamada passando o valor 0 (ou seja, $f(0)$), o retorno da execução será o valor -1.



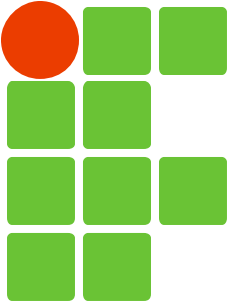
INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Diagnóstico

```
struct aluno {  
    float nota;  
    int matricula;  
}
```

- a) O trecho de código define uma variável cujo nome é aluno.
- b) O trecho define uma estrutura composta por dois campos, um do tipo float e outro do tipo int.
- c) O trecho de código define duas variáveis, uma do tipo float e outra do tipo int.
- d) O trecho de código define um vetor de duas posições.



INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Diagnóstico

```
typedef struct Aluno{  
    float nota;  
    int matricula;  
} TAluno;
```

```
int melhor(TAluno a1, TAluno a2){  
    if (a1.nota > a2.nota)  
        return a1.matricula;  
    else  
        return a2.matricula;  
}
```

a) A função “melhor” apresenta erro de sintaxe. Nos lugares onde se utilizou ponto(“.”) deveria se utilizar (“->”). Por exemplo, onde está “a1.nota” deveria ser “a1->nota”.

b) A função “melhor” não apresenta erros de sintaxe. Ela recebe os dados de dois alunos e retorna a matricula do aluno com maior nota.

c) A função “melhor” não apresenta erros de sintaxe. Ela recebe os dados de dois alunos e retorna os dados do aluno com maior nota.

d) A função “melhor” apresenta erro de sintaxe: como os parâmetros de entrada são do tipo TAluno, o tipo de retorno deveria ser TAluno e não int.