

Roteiro Aula Prática



**ALGORITMOS E TÉCNICAS
DE PROGRAMAÇÃO**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: ALGORITMOS E TÉCNICAS DE PROGRAMAÇÃO

Unidade: 2 – CONSTANTES, VARIÁVEIS E OPERAÇÕES

Aula: 3 – OPERAÇÕES E EXPRESSÕES

OBJETIVOS

- ✓ Compreender a definição e utilização de constantes e variáveis em linguagens de programação.
- ✓ Familiarizar-se com tipos de dados primitivos e compostos.
- ✓ Praticar operações e expressões em C.
- ✓ Explorar o uso de ponteiros e a manipulação de memória.

SOLUÇÃO DIGITAL

- Computador com um compilador C/C++ instalado juntamente com a IDE Code::Blocks.

LINK SOLUÇÃO DIGITAL (EXCETO ALGETEC): <https://www.codeblocks.org/downloads/>

Code::Blocks é um ambiente de desenvolvimento integrado de código aberto e multiplataforma. Ele foi desenvolvido em C++, usando wxWidgets. Sua arquitetura é orientada a plugin, de forma que suas funcionalidades são definidas pelos plugins fornecidos a ele. Code::Blocks é voltado para o desenvolvimento em C/C++.

PROCEDIMENTO/ATIVIDADE

ATIVIDADE PROPOSTA:

Desenvolver um programa em C que utilize variáveis primárias e compostas, ponteiros, constantes e operações matemáticas. Bem como funções de exibição de resultados.

PROCEDIMENTOS PARA A REALIZAÇÃO DA ATIVIDADE:

Passo 1: Incluindo Bibliotecas e Definindo Constantes

Começamos incluindo as bibliotecas necessárias para funções de entrada/saída (`stdio.h`) e manipulação de strings (`string.h`). Em seguida, definimos algumas constantes.

```
#include <stdio.h>
```

```
#include <string.h>
```

```
// Definição de constantes
#define MAX_STRING_LENGTH 50
const int CONST_NUM = 100;
```

Passo 2: Declaração de Variáveis

Declaramos as variáveis que usaremos no programa, incluindo variáveis primárias (inteiros, float, string), variáveis compostas (vetor, matriz) e ponteiros.

```
int main() {
    // Variáveis primárias
    int num1, num2;
    float num3;
    char str[MAX_STRING_LENGTH];

    // Variáveis compostas
    int vetor[3] = {1, 2, 3}; // Vetor inicializado com 3 elementos
    int matriz[2][2] = {{1, 2}, {3, 4}}; // Matriz 2x2 inicializada

    // Ponteiros
    int *ptr_num1;
```

Passo 3: Atribuição Inicial

Fazemos a atribuição inicial de valores para algumas variáveis e atribuímos o endereço de `num1` ao ponteiro `ptr_num1`.

```
// Atribuição inicial
num1 = 10;
num2 = 20;
num3 = 15.5;
strcpy(str, "Exemplo");

// Atribuição de ponteiro
ptr_num1 = &num1;
```

Passo 4: Entrada do Usuário

Solicitamos ao usuário que insira valores para `num1`, `num2`, `num3` e `str`.

```
// Entrada do usuário
printf("Digite um número inteiro: ");
scanf("%d", &num1);

printf("Digite outro número inteiro: ");
scanf("%d", &num2);

printf("Digite um número real: ");
scanf("%f", &num3);

printf("Digite uma string: ");
scanf("%s", str);
```

Passo 5: Operações Matemáticas e Booleanas

Realizamos algumas operações matemáticas (soma e média) e booleanas, e armazenamos os resultados em variáveis.

```
// Operações matemáticas
int soma = num1 + num2;
float media = (num1 + num2 + num3) / 3;

// Operações booleanas
int resultado_bool = (num1 > num2) && (num3 > CONST_NUM);
```

Passo 6: Exibição dos Resultados

Exibimos os resultados das operações matemáticas e booleanas, além dos valores dos elementos do vetor, matriz e ponteiro, e os endereços de memória.

```
// Exibição dos resultados
printf("Soma de num1 e num2: %d\n", soma);
printf("Média dos números: %.2f\n", media);
printf("Resultado da operação booleana: %d\n", resultado_bool);

// Exibição dos elementos do vetor
printf("Elementos do vetor: %d, %d, %d\n", vetor[0], vetor[1], vetor[2]);

// Exibição dos elementos da matriz
```

```
printf("Elementos da matriz: %d, %d, %d, %d\n", matriz[0][0], matriz[0][1], matriz[1][0],  
matriz[1][1]);
```

```
// Manipulação e exibição do ponteiro  
printf("Valor de num1: %d\n", *ptr_num1);  
printf("Endereço de num1: %p\n", ptr_num1);  
printf("Endereço de str: %p\n", (void*)&str);  
  
return 0;  
}
```

Programa Completo:

```
#include <stdio.h>  
#include <string.h>  
  
// Definição de constantes  
#define MAX_STRING_LENGTH 50  
const int CONST_NUM = 100;  
  
int main() {  
    // Variáveis primárias  
    int num1, num2;  
    float num3;  
    char str[MAX_STRING_LENGTH];  
  
    // Variáveis compostas  
    int vetor[3] = {1, 2, 3}; // Vetor inicializado com 3 elementos  
    int matriz[2][2] = {{1, 2}, {3, 4}}; // Matriz 2x2 inicializada  
  
    // Ponteiros  
    int *ptr_num1;  
  
    // Atribuição inicial  
    num1 = 10;  
    num2 = 20;  
    num3 = 15.5;  
    strcpy(str, "Exemplo");
```

```
// Atribuição de ponteiro
ptr_num1 = &num1;

// Entrada do usuário
printf("Digite um número inteiro: ");
scanf("%d", &num1);

printf("Digite outro número inteiro: ");
scanf("%d", &num2);

printf("Digite um número real: ");
scanf("%f", &num3);

printf("Digite uma string: ");
scanf("%s", str);

// Operações matemáticas
int soma = num1 + num2;
float media = (num1 + num2 + num3) / 3;

// Operações booleanas
int resultado_bool = (num1 > num2) && (num3 > CONST_NUM);

// Exibição dos resultados
printf("Soma de num1 e num2: %d\n", soma);
printf("Média dos números: %.2f\n", media);
printf("Resultado da operação booleana: %d\n", resultado_bool);

// Exibição dos elementos do vetor
printf("Elementos do vetor: %d, %d, %d\n", vetor[0], vetor[1], vetor[2]);

// Exibição dos elementos da matriz
printf("Elementos da matriz: %d, %d, %d, %d\n", matriz[0][0], matriz[0][1], matriz[1][0],
matriz[1][1]);

// Manipulação e exibição do ponteiro
printf("Valor de num1: %d\n", *ptr_num1);
```

```
printf("Endereço de num1: %p\n", ptr_num1);
printf("Endereço de str: %p\n", (void*)&str);

return 0;
}
```

Para verificar o seu entendimento com relação aos conceitos de variáveis e operações básicas em C, desenvolva um pequeno programa em C que realize as seguintes tarefas:

1. Defina uma constante que represente a taxa de juros anual (por exemplo, 5%).
2. Declare variáveis para armazenar o valor inicial do investimento, o número de anos e o valor final do investimento.
3. Solicite ao usuário que insira o valor inicial do investimento e o número de anos.
4. Calcule o valor final do investimento utilizando a fórmula de juros simples: $\text{Valor Final} = \text{Valor Inicial} * (1 + (\text{Taxa de Juros} * \text{Anos}))$.
5. Exiba o valor final do investimento.
6. Além disso, declare um vetor de inteiros com 3 elementos e uma matriz 2x2 de inteiros. Inicialize-os com valores à sua escolha.
7. Utilize um ponteiro para acessar e modificar o primeiro elemento do vetor.
8. Exiba os valores dos elementos do vetor e da matriz, bem como os endereços de memória de cada elemento.

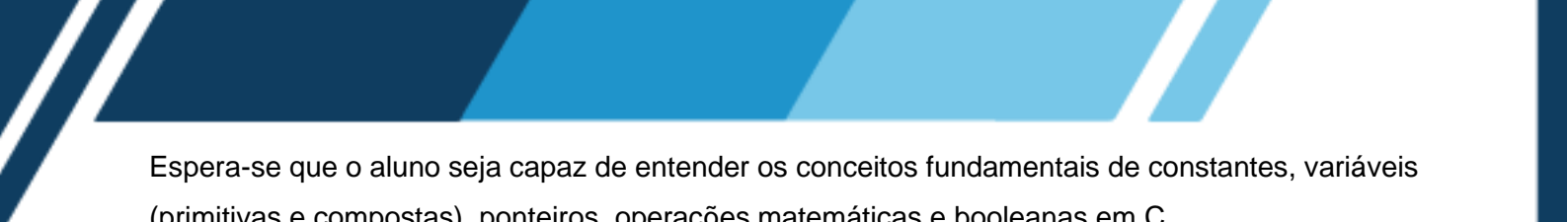
CHECKLIST:

- Incluindo Bibliotecas e Definindo Constantes
- Declaração de Variáveis
- Atribuição Inicial
- Entrada do Usuário
- Operações Matemáticas e Booleanas
- Exibição dos Resultados
- Resolução do exercício proposto

RESULTADOS

Para comprovar a realização da atividade, é necessário entregar um arquivo .c que contém o seu código da atividade.

RESULTADOS DE APRENDIZAGEM:



Espera-se que o aluno seja capaz de entender os conceitos fundamentais de constantes, variáveis (primitivas e compostas), ponteiros, operações matemáticas e booleanas em C.