Algoritmos e Programação II

Prof.ª Noeli A. Pimentel Vaz
Prof. Joilson dos Reis Brito

Roteiro da Aula

- ✔ Correção de Exercício - Funções
- ✔ Passagem de Parâmetros por valor
- ✔ Ponteiros
- ✔ Passagem de Parâmetros por referência

Questão 6

Faça uma função chamada MULTIPLO que recebe como parâmetro dois números inteiros e retorna o 1, se o primeiro número for múltiplo do segundo, número e o 0 caso contrário.
Exemplo de utilização da função:

MULTIPLO(35,5)
Retornará o valor 1

MULTIPLO(35,2)
Retornará o valor 0

Depois faça um programa que lê dois números inteiros e escreve se o primeiro é múltiplo do segundo, utilizando a função MULTIPLO.

Resposta - Questão 6

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

int multiplo(int Numero1, int Numero2)
{
   if(Numero1 % Numero2 == 0)
      return 1;
   else
      return 0;
}
```

Resposta - Questão 6

```
int main()
{
    int Valor1, Valor2, Retorno;
     setlocale(LC_ALL, "Portuguese");
    printf("Digite dois números inteiros: ");
    scanf("%d%d",&Valor1, &Valor2);
    Retorno = multiplo(Valor1, Valor2);
    if(Retorno)// é o mesmo que Retorno == 1
      printf("\n%d é multiplo de %d.\n",Valor1, Valor2);
    else
      printf("\n%d não é multiplo de %d.\n",Valor1, Valor2);
    return 0;
}
```

Questão 10

Faça um programa que implementa uma calculadora que faz as 4 operações. No programa devem ser implementadas 4 funções: soma, subtrai, multiplica e divide. Cada função deve receber dois números reais como parâmetro e retornar um número real que será o resultado da operação feita sobre os dois números passados como parâmetro.

O programa principal deve ler dois números e uma operação(+ - * /), chamar a função que faz a operação matemática informada e guardar o retorno da função em uma variável que armazena o resultado atual da calculadora.

Depois o programa deve entrar em um laço pedindo para o usuário informar uma operação e um número , chamar a função referente a operação, passando como parâmetro a variável que tem o resultado atual da calculadora e o número lido, depois colocar o retorno da função na variável que armazena o resultado atual da calculadora.

A cada operação feita o programa deverá mostrar o valor da variável que armazena o resultado atual da calculadora.

O programa deve sair do laço quando o usuário digitar F na operação.

Se o usuário digitar uma operação diferente de +, -, *,/ e F o programa deve informar que o operação é inválida.

Resposta - Questão 10

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

float soma(float Valor1, float Valor2)
   return Valor1+Valor2;

float subtrai(float Valor1, float Valor2)
  return Valor1-Valor2;


float multiplica(float Valor1, float Valor2)
  return Valor1*Valor2;

float divide(float Valor1, float Valor2)
  return Valor1/Valor2;
```

Resposta - Questão 10

```
int main()
{
    float Numero1, Numero2, Resultado;
    char Operacao;
    setlocale(LC_ALL,"Portuguese");
    printf("\t\tC A L C U L A D O R A \n");
    printf("\nDigite o primeiro número: ");
    scanf("%f",&Numero1);
   printf("Digite a operação (+, -, /, *): ");
    scanf("%c",&Operacao);
    printf("Digite o segundo número: ");
    scanf("%f",&Numero2);
    switch(Operacao)
    {
        case '+': Resultado = soma(Numero1, Numero2);
                  break;
        case '-': Resultado = subtrai(Numero1, Numero2);
                  break;
        case '*': Resultado = multiplica(Numero1, Numero2);
                  break;
        case '/': Resultado = divide(Numero1, Numero2);
                  break;
    }
```

Resposta - Questão 10

```
    do
    {
          system("cls");
          printf("\t\tC A L C U L A D O R A \n");
          printf("\nResultado atual: %.2f\n\n",Resultado);
          printf("\n\nDigite a operação: ");
          fflush(stdin);
          scanf("%c",&Operacao);
          if(Operacao == 'F')
            break;
          printf("Digite o valor: ");
          scanf("%f",&Numero2);
          switch(Operacao)
          {
            case '+': Resultado = soma(Resultado, Numero2);
                                  break;
            case '-': Resultado = subtrai(Resultado, Numero2);
                                 break;
            case '*': Resultado = multiplica(Resultado, Numero2);
                                 break;
            case '/': Resultado = divide(Resultado, Numero2);
                                 break;
            case 'F': break;
            default: printf("Operação inválida!");
                        system(“Pause”);
          }
    }while(Operacao != 'F');
    return 0;
}
```

Questão 11

Faça um programa que simule a urna eletrônica para eleição de síndico de um prédio. Seu programa deve ter uma função chamada **urna** que deve apresentar as opções de votação, ler a opção, aceitando apenas valores de 1 a 5, e retornar o valor digitado pelo usuário.

ELEIÇÕES PARA SÍNDICO
---
1. Joaquim Manoel Andrade
2. Mariana Castro
3. Nulo
4. Branco
5. Relatório

Digite o voto:

O programa deverá chamar a função urna até que seja digitado o número 5. No final o programa deverá apresentar as seguintes informações:
Total de votos registrados
Quantidade de votos de cada candidato;
A porcentagem de votos nulos;
A porcentagem de votos brancos;
O candidato vencedor.

Resposta - Questão 11

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

int urna()
{
	int Voto;
	system("cls");
	printf("--------------------------------------------------\n");
	printf("\t\tE L E I Ç Õ E S   2 0 20 - 2o Turno\n");
	printf("--------------------------------------------------\n");
	printf("\t\t\t1. Joaquim Manoel Andrade\n");
	printf("\t\t\t2. Mariana Castro\n");
	printf("\t\t\t3. Nulo\n");
	printf("\t\t\t4. Branco\n");
	printf("\t\t\t5. Relatório\n");
	printf("\t\t--------------------------------------------------\n");
	printf("\t\tDigite o voto:");
	scanf("%d",&Voto);
	while(Voto < 1 || Voto > 5)
	{
	   printf("Opção inválida! Digite novamente.\n");
	   scanf("%d",&Voto);
	}
	return Voto;
}
```

Resposta - Questão 11

```
int main()
{
        int Opcao,
       float Votos1 = 0, Votos2 = 0, VotosNulo = 0, VotosBranco = 0,TotalVotos;;
        setlocale(LC_ALL,"Portuguese");
        do
        {
         Opcao = urna();
         switch (Opcao)
         {
                case 1:
                      Votos1++;
                      break;
                case 2:
                      Votos2++;
                      break;
                case 3:
                      VotosNulo++;
                      break;
                case 4:
                      VotosBranco++;
                      break;
                case 5:
                      system("cls");
                      printf("** RELATÓRIO **:");
                      printf("\nTotal Roberto: %.f",Votos1);
                      printf("\nTotal Antônio: %.f",Votos2);
                      printf("\nTotal Nulo: %.f",VotosNulo);
                      printf("\nTotal Branco: %.f",VotosBranco);
                    TotalVotos= Votos1+Votos2+VotosBranco+VotosNulo;
                      printf("\nPorcentagem Votos Nulos: %.1f",(VotosNulo/TotalVotos)*100);
                      printf("\nPorcentagem Votos Brancos: %.1f",(VotosBranco/TotalVotos)*100);
                      if(Votos1>Votos2)
                        printf("\nJoaquim Manoel Andrade é o vencedor!");
                      else
                       if(Votos1<Votos2)
                           printf("\nMariana Castro é a vencedora!");
                       else
                           printf("\nEmpate");
                      break;
         }
        }while(Opcao != 5);
        printf("\n\nFIM DAS ELEIÇÕES.\n");
        return 0;
}
```

Questão 9

Faça um programa que solicita o total gasto pelo cliente de uma loja, imprime as opções de pagamento, solicita a opção desejada e imprime o valor total que o cliente irá pagar.

1) A vista com 10% de desconto
2) 2 vezes (preço da etiqueta)
3) 3 vezes com 3% de juros ao mês (somente para compras acima de R$ 100,00).

OBS: fazer uma função que imprime as opções solicita a opção desejada e retorna a opção escolhida. No programa principal, testar a opção escolhida e ativar a função correspondente (uma função para cada opção).

Resposta - Questão 9

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

int leropcao()
{
	int Op;
	system("cls");
	printf("Digite a opção desejada para o pagamendo: \n");
	printf("1) A vista com 10%% de desconto\n");
	printf("2) 2 vezes (preço da etiqueta)\n");
	printf("3) 3 vezes com 3%% de juros ao mes (somente para compras acima de R$ 100,00).\n");
	scanf("%i",&Op);
	return Op;
}
```

Resposta - Questão 9

```
void funcao1(float Valor)
{
   printf("Valor a vista com 10%% de desconto: %.2f\n",Valor*0.9);
}

void funcao2(float Valor)
{
     printf("Duas vezes de: %.2f\n",Valor/2);
}

void funcao3(float Valor)
{
     if(Valor >= 100)
     {
        printf("Três vezes de: %.2f\n",((Valor/3)*1.03));
     }
     else
       printf("Não é permitido parcelar em 3 vezes abaixo de 100,00\n");
}
```

Resposta - Questão 9

```
int main()
{
	int Opcao;
	float TotalGasto;
	setlocale(LC_ALL,"Portuguese");
	printf("Digite o total gasto na compra: ");
	scanf("%f",&TotalGasto);
	Opcao = leropcao();
	while(Opcao < 1 || Opcao > 3)
	{
		printf("Opção inválida! Digite novamente!");
		system("Pause");
		Opcao = leropcao();
	}
	switch (Opcao)
	{
		case 1: funcao1(TotalGasto);
			system("Pause");
			break;
		case 2: funcao2(TotalGasto);
			system("Pause");
			break;
		case 3: funcao3(TotalGasto);
			system("Pause");
			break;
	}
	return 0;
}
```

Passagem de Parâmetros para Função

Quando realizamos a ação de enviar um ou vários valores para que a função utilize, estamos realizando a **passagem parâmetros.**

A passagem de parâmetros pode ser realizada de duas maneiras:

- Por Valor
- Por Referência

Passagem de Parâmetros **por Valor**

A passagem de parâmetros por valor é o tipo de passagem de parâmetros que estamos utilizando até agora nos programas com funções.

Na passagem de parâmetros por valor, fazemos uma cópia do conteúdo da variável de origem (função main) para a variável declarada entre os parênteses na função.

Se houver modificação no parâmetro dentro da função, esta alteração não é realizada na variável de origem, pois os valores somente foram copiados.

Passagem de Parâmetros **por Valor**

Exemplo:

```
...
float soma(float Valor1, float Valor2)
{
        return Valor1+Valor2;
}
…
int main()
{
        float Numero1, Numero2, Resultado;
        char Operacao;
        setlocale(LC_ALL,"Portuguese");
        printf("\t\tC A L C U L A D O R A \n");
        printf("\nDigite o primeiro número: ");
        scanf("%f",&Numero1);
       printf("Digite a operação (+, -, /, *): ");
        fflush(stdin);
        scanf("%c",&Operacao);
        printf("Digite o segundo número: ");
        scanf("%f",&Numero2);
        switch(Operacao)
        {
                case '+': Resultado = soma(Numero1, Numero2);
                break;
        ...
```

Qualquer modificação, caso aconteça, com as variáveis Valor1 e Valor2 não será visualizada fora da função soma.

Quando a função soma é ativada, o conteúdo de Numero1 é atribuído para Valor1 e o conteúdo de Numero2 é atribuído para Valor2, nesta ordem.

Passagem de Parâmetros **por Referência**

A passagem de parâmetros por referência é o tipo de passagem de parâmetros que utiliza Ponteiros para sua implementação.

Na passagem de parâmetros por referência, enviamos a referência da variável de origem (função main) para a variável declarada entre os parênteses na função, ou seja, o endereço no qual a variável está armazenada na memória RAM.

Se houver modificação no parâmetro dentro da função, esta alteração é realizada na variável de origem, pois as duas variáveis acessam o mesmo endereço de memória.

Passagem de Parâmetros **por Referência**

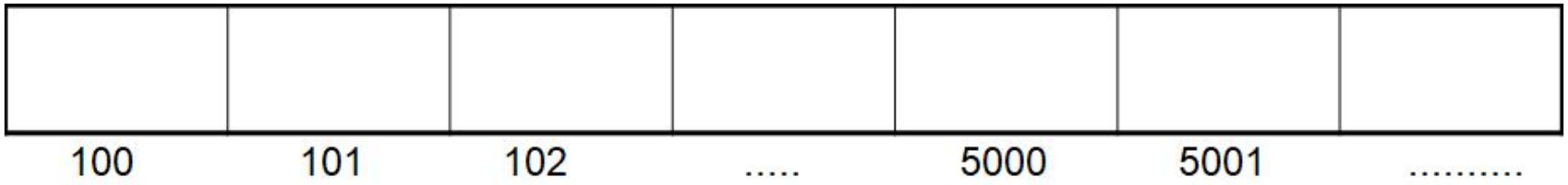
Para entendermos como funciona a Passagem de Parâmetros por Referência, vamos agora estudar **PONTEIROS**!

PONTEIROS

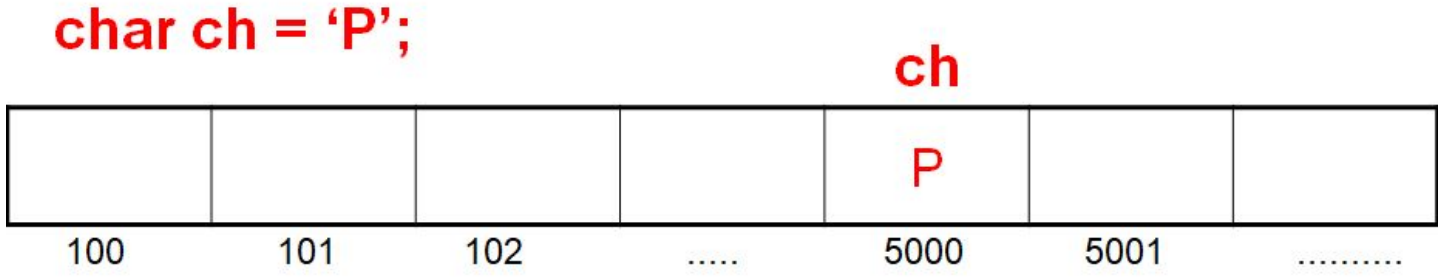
A Linguagem C permite que o programador referencie a posição dos dados bem como os próprios dados (isto é, o conteúdo de suas posições).

Um ponteiro é um **endereço** de memória que pode ser acessado.

A memória de qualquer computador é uma sequência de bytes:



Sempre que declaramos uma variável, temos que indicar qual seu tipo e nome:



Ponteiros

Portanto para cada variável que definimos temos as seguintes informações:

Nome

Endereço de memória da variável

Conteúdo da variável, ou do seu endereço de memória.

Um ponteiro é uma variável que armazena o endereço de memória de outra variável.

Ponteiros

O ponteiro é uma variável como outra qualquer, porém seu conteúdo será um endereço de memória, por isso deverá ser declarado:

Sintaxe: tipo *ptr;

ptr – É o nome da variável do tipo ponteiro;

tipo – É o tipo da variável para a qual apontará;

* - Indica que é uma variável do tipo ponteiro;

Ponteiros

Exemplo:

```
int *pi;

float *pf;

char *pc;
```

pi é um ponteiro para inteiro;

pf é um ponteiro para um real;

char é um ponteiro para um char;

O asterisco indica que valores das variáveis declaradas são ponteiros para valores dos tipos especificados na declaração.

Ponteiros

Se x for declarado como um inteiro, &x se refere ao endereço de memória reservado para armazenar o conteúdo da variável x.

&x é chamado PONTEIRO.

Por isso usamos & para realizar entrada de dados com a função scanf. O que estamos indicando no comando é que o valor será armazenado no endereço (&) de memória da variável.

Ponteiros

Um ponteiro é como qualquer outro tipo de dado em C.

O valor de um ponteiro é uma posição de memória, da mesma forma que o valor de um inteiro é um número.

Os valores dos ponteiros podem ser atribuídos.

Exemplo:

```
int x, *pi;
pi = &x;
```

Ponteiros

Usar um asterisco antes de uma variável ponteiro é a maneira que a linguagem C utiliza para acessar o conteúdo armazenado no endereço do ponteiro, ou seja, o valor armazenado no endereço apontado pelo ponteiro.

Exemplo:

```
int *PonteiroNumero, Numero1, Numero2;

Numero1 = 10;

PonteiroNumero = &Numero1;

Numero2 = *PonteiroNumero + 1;
```

Atribui a Numero2 o valor de Numero1 +1, utilizando o ponteiro.

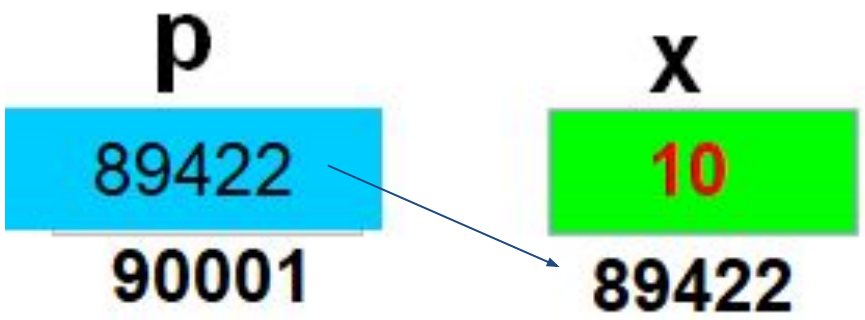
Ponteiros

Exemplo:

```
int *p, x, y;

x = 10;
p = &x;
```

*p é 10



Exemplo - Programa que usa Ponteiros

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

int main()
{
  int *p, *q, a, b, c;
  setlocale(LC_ALL,"Portuguese");
  printf("Digite dois numeros: \n");
  scanf("%d %d", &a, &b);
  p = &a;
  q = &b;
  c = *p + *q;

  printf("Variável a: %d \n Endereço de memória: %p \n", a, p);
  printf("\n\nVariável b: %d \n Endereço de memória:   %p \n", b, q);
  printf("%i + %i: %i \n",*p, *q, c);
  system("PAUSE");
  return 0;
}
```

Ponteiros

Aprendi ponteiros!

E agora? Onde vou usá-los?

Ponteiros são usados para:

- **Passagem de Parâmetros por Referência**
- **Alocação dinâmica de Memória**

Passagem de Parâmetros por Referência

Acabamos de ver que:

- Na passagem de **parâmetros por referência**, enviamos a referência (**endereço**) da variável de origem (função main) para a variável declarada entre os parênteses na função, ou seja, o endereço que a variável está armazenada na memória RAM.
- Se houver modificação no parâmetro dentro da função, esta alteração é realizada na variável de origem, pois as duas variáveis acessam o mesmo endereço de memória.

USAREMOS OS PONTEIROS PARA ISSO!

Exemplo 1 - Passagem de Parâmetros por Referência

Programa que faz a leitura de dois valores inteiros e efetua a troca do conteúdo das variáveis dentro de uma função.

Como uma função não pode retornar mais que um valor, a passagem de parâmetros por referência é usada para modificar o conteúdo de duas variáveis dentro da função utilizando ponteiros.

Exemplo 1 - Passagem de Parâmetros por Referência

Programa que faz a leitura de dois valores inteiros e efetua a troca do conteúdo das variáveis dentro de uma função.

```
#include<stdio.h>
#include<stdlib.h>
#include<locale.h>
void troca(int *PonteiroA, int *PonteiroB)
{
        int Aux;
        Aux = *PonteiroA;
        *PonteiroA = *PonteiroB;
        *PonteiroB = Aux;
}
//FUNÇÃO PRINCIPAL
int main()
{
        int ValorA, ValorB;
        setlocale(LC_ALL,"Portuguese");
        printf("Digite dois números inteiros: ");
        scanf("%d%d",&ValorA,&ValorB);
        troca(&ValorA, &ValorB);
        printf("Valores após passagem de parâmetro por Referência\n");
        printf("ValorA: %d - ValorB: %d\n", ValorA,ValorB);
        system("Pause");
        return 0;
}
```