



# Algoritmos e Programação II

Prof. Joilson dos Reis Brito

# Roteiro da Aula

- ✓ Correção Lista
- ✓ Passagem de Parâmetros na Linha de Comando
- ✓ Os parâmetros da função `main(int argc , char * argv[ ])`
- ✓ Correção Prova 1a VA

# Roteiro da Aula

- ✓ Correção Lista
- ✓ Passagem de Parâmetros na Linha de Comando
- ✓ Os parâmetros da função `main(int argc , char * argv[ ])`
- ✓ Correção Prova 1a VA

## 2 - Analise o código fonte:

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

```
int main()  
{  
    int ValorA, ValorB;  
    printf("Digite dois numeros inteiros: ");  
    scanf("%d%d",&ValorA, &ValorB);  
    verificavalor(&ValorA,&ValorB);  
    printf("Maior valor: %d\n",ValorA);  
    printf("Menor valor: %d\n",ValorB);  
    return 0;  
}
```

**Desenvolva a função chamada verificavalor, para que o programa funcione corretamente:**

```
void verificavalor(int *Numero1, int *Numero2)  
{  
  int Auxiliar;  
  if (*Numero1 < *Numero2)  
  {  
    Auxiliar = *Numero1;  
    *Numero1 = *Numero2;  
    *Numero2 = Auxiliar;  
  }  
}
```

3 - Analise o código fonte:

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

```
void somavalores(float *A, float *B)
```

```
{
```

```
    *A = *A + *B;
```

```
}
```

Desenvolva a função principal para que o programa realize a leitura de dois números, chame a função **somavalores** e imprima a soma dos números.

```
int main()  
{  
    float ValorA, ValorB;  
    printf("Digite o 1o Numero: ");  
    scanf("%f",&ValorA);  
    printf("Digite o 2o Numero: ");  
    scanf("%f",&ValorB);  
    somavalores(&ValorA, &ValorB);  
    printf("Soma dos Numeros: %.2f\n",ValorA);  
}
```

**5 - Desenvolva uma função que calcule a área da superfície esférica e o volume de uma esfera de raio  $R$ .**

**Essa função deve obedecer ao protótipo:**

**`void calcesfera(float R, float *Area, float *Volume)`**

**Sabendo-se que**



```
#include<stdio.h>
#include<math.h>
void calcesfera(float R, float *Area, float *Volume)
{
    *Area = 4*M_PI *pow(R,2);
    *Volume = 4/3*M_PI*pow(R,3);
}

int main()
{
    float Raio, AreaEsfera, VolumeEsfera;
    printf("Digite o raio da esfera: ");
    scanf("%f",&Raio);
    calcesfera(Raio,&AreaEsfera,&VolumeEsfera);
    printf("Area da Esfera: %.2f\n",AreaEsfera);
    printf("Volume da Esfera: %.2f\n",VolumeEsfera);
}
```

**6 - Faça um programa com funções para:**

- Preencher um vetor Numeros, de 15 posições inteiras, com números aleatórios de 0 a 100.**
- Imprimir o vetor gerado.**
- Retornar a quantidade de valores pares que existem no vetor números.**

**Depois faça a função principal que chama as funções e imprime quantos números pares o vetor números tem.**

```
#include<stdio.h>  
#include<stdlib.h>  
#include<time.h>  
#include<locale.h>  
#define TAMANHO 15  
  
void preenche(int *VetorNumeros)  
{  
    int l;  
    srand(time(NULL));  
    for(l=0;l<TAMANHO;l++)  
        VetorNumeros[l]=rand()%100;  
}
```

```
void imprime(int *VetorNumeros)  
{  
    int l;  
    printf("Vetor Gerado\n");  
    for(l=0;l<TAMANHO;l++)  
        printf("%2d ",VetorNumeros[l]);  
    printf("\n");  
    system("pause");  
}  
int contapares(int *VetorNumeros)  
{  
    int l,Contador=0;  
    for(l=0;l<TAMANHO;l++)  
        if(VetorNumeros[l] % 2 ==0)  
            Contador ++;  
    return Contador;  
}
```

```
int main()  
{  
    setlocale(LC_ALL, "Portuguese");  
    int Numeros[TAMANHO];  
    preenche(Numeros);  
    imprime(Numeros);  
    printf("O Vetor tem %d números pares?\n",contapares(Numeros));  
    system("pause");  
}
```

# Passagem de Parâmetros na Linha de Comando

Quase todos os usuários de computadores conhecem utilitários ou programas próprios do sistema operacional que permitem que sejam passados parâmetros na linha de comando na qual o executável é executado.

No Prompt de Comando, para copiar o arquivo alfa para o arquivo beta bastaria executar o comando copy com o nome do arquivo origem e do arquivo destino.

```
C:\> copy alfa.txt beta.txt
```

**Pergunta:** Quem é o destinatário dos argumentos passados na linha de comando?

**Resposta:** Só pode ser o próprio programa copy, pois é ele que tem que saber qual o nome do arquivo origem e destino.

# Passagem de Parâmetros na Linha de Comando

**Pergunta:** Como será então possível receber, dentro de um programa, os argumentos(parâmetros) passados na linha de comando?

**Resposta:** Quando o programa inicia a sua execução já deverá conhecer qual a linha de comando que lhe foi enviada. Por isso, os argumentos da linha de comando deverão ser passados à primeira função a ser executada num programa em C, a **função main**.

# PASSANDO PARÂMETROS PARA UM PROGRAMA

Assim, a função `main` pode receber os parâmetros que foram passados na linha de comando através de dois parâmetros:

```
main(int argc, char *argv[])
```

em que:

**argc** — é um inteiro que indica quantos argumentos foram passados na linha de comando (incluindo o próprio nome do programa).

**argv** — vetor contendo todas as strings passadas na linha de comando.

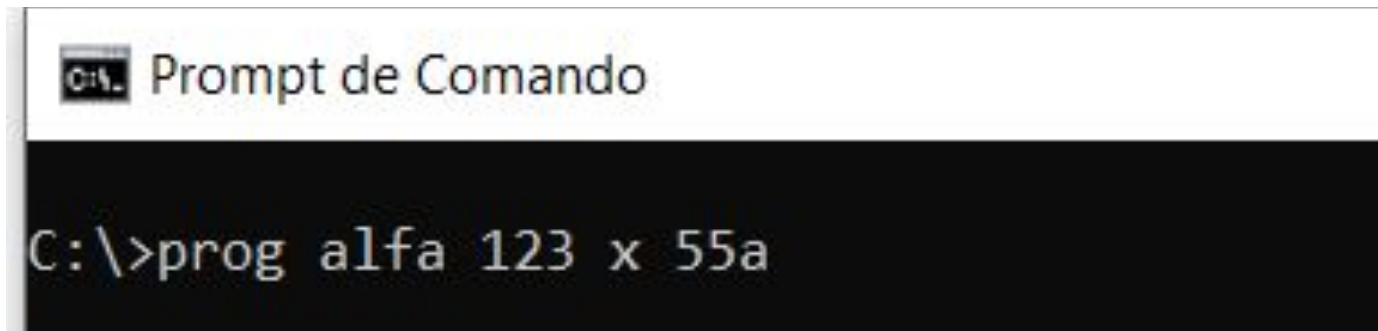
Note que `argv` é um vetor, pois está declarado como `argv[]`. Os elementos que armazena são do tipo `char *` (strings).

Portanto, `argv` é um vetor de strings.



# PASSANDO PARÂMETROS PARA UM PROGRAMA

Suponhamos, então, que estamos escrevendo um programa denominado prog e lhe passamos os seguintes parâmetros: alfa 123 x 55a.



```
C:\>prog alfa 123 x 55a
```

O valor de argc será 5 (e não 4, pois inclui o nome do executável).

O vetor argv irá conter as strings passadas na linha de comando.

Observe que todos os parâmetros são armazenados como strings (valores numéricos incluídos).

# PASSANDO PARÂMETROS PARA UM PROGRAMA

argv[0]	...	p	r	o	g	\0
argv[1]	...	a	l	f	a	\0
argv[2]	...	1	2	3	\0	
argv[3]	...	x	\0			
argv[4]	...	5	5	a	\0	

## Programa que mostra os parâmetros passados ao programa

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

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

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

```
int main(int argc, char *argv[])
```

```
{
```

```
    int i;
```

```
    system("cls");
```

```
    printf("Quantidade de parametros (Argc): %d\n",argc);
```

```
    for( i=0;i<argc;i++)
```

```
    {
```

```
        printf("\nParametro argv[%d]: %s\n",i,argv[i]);
```

```
    }
```

```
    system("Pause");
```

```
    return 0;
```

```
}
```

## EXEMPLO UTILIZANDO O PROMPT DE COMANDOS DO WINDOWS

Para executar o programa no prompt navegue até a pasta onde você salvou o arquivo Parametros.cpp (utilizando o comando cd) e digite:

### >Parametros

```
C:\Users\Lenovo\Documents\LP11>Parametros
Quantidade de parametros (Argc): 1

Parametro argv[0]: Parametros
Pressione qualquer tecla para continuar. . . .
```

## EXEMPLO UTILIZANDO O PROMPT DE COMANDOS DO WINDOWS

Para executar o programa no prompt navegue até a pasta onde você salvou o arquivo Parametros.cpp (utilizando o comando cd) e digite:

>Parametros Valor1

```
C:\Users\Lenovo\Documents\LP11>Parametros Valor1
Quantidade de parametros (Argc): 2

Parametro argv[0]: Parametros

Parametro argv[1]: Valor1
Pressione qualquer tecla para continuar
```

## EXEMPLO UTILIZANDO O PROMPT DE COMANDOS DO WINDOWS

Programa que recebe uma palavra por parâmetro e escreve e escreve esta palavra da esquerda para direita

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define TAMANHO 100
int main(int argc, char *argv[])
{
    system("cls");
    int i;
    char Palavra[TAMANHO];
    strcpy(Palavra,argv[1]);
    for(i=strlen(Palavra)-1;i>=0;i--)
        printf("%c",Palavra[i]);
    printf("\n");
    system("Pause");
    return 0;
}
```

## EXEMPLO UTILIZANDO O PROMPT DE COMANDOS DO WINDOWS

**Programa que recebe vários números e escreve o maior**

```
#include<stdlib.h>
#include<time.h>
int main(int argc, char *argv[])
{
    system("cls");
    int Maior,l,Numero;
    Maior = atoi(argv[1]);
    for(l=2; l<=argc;l++)
    {
        Numero =atoi(argv[l]);
        if(Numero > Maior)
            Maior = Numero;
    }
    printf("Maior valor: %d\n",Maior);
    system("pause");
}
```