

Engenharia de Computação – Campus Nova Gameleira  
Disciplina: Programação de Computadores I (PC I) – Turma Extra - Semestre: 1º/2023  
Prof.: Evandrino Barros (evandrino@cefetmg.br)

Exercício 4 – Funções e Procedimentos - em duplas, 2 pontos – entrega somente pelo AVA.

Gerar um arquivo PDF com os códigos fontes em C dos programas solicitados abaixo, incluindo as telas de execução de cada um. Utilize os recursos do seu computador para imprimir o resultado das telas (*print screen*). Todos os recursos da Linguagem C para solução desse exercício são apresentados nas transparências de PC I.

A entrega deve ser feita pelo AVA da disciplina (<https://ava.cefetmg.br/course/view.php?id=879>) no link disponível para isso. Envie somente um PDF por dupla. Inclua o nome de cada integrante no documento. Inclusões posteriores não serão aceitas.

1) Crie um programa para imprimir, por meio de duas repetições separadas, uma linha de 5 asteriscos e uma outra linha de 10 asteriscos. A cada iteração das repetições, somente um asterisco deve ser impresso por vez. Crie o programa prog1.c.

2) Crie o prog2.c a partir do prog1.c anterior, sendo que agora o prog2.c deve ter as funções linha5 e linha10 para fazer as mesmas impressões feitas pelo prog1.c. Novamente, imprimir um único asterisco a cada iteração nas repetições usadas.

3) Crie o prog3.c a partir do prog2 anterior, sendo que agora este programa deve ter somente uma função de nome linha para imprimir as linhas de 5 e 10 asteriscos. A função deve ter um parâmetro para receber a quantidade de asteriscos. Novamente, imprimir um único asterisco a cada iteração nas repetições usadas.

4) Crie o prog4.c, a partir do prog3.c anterior, adicionando um parâmetro para o caractere a ser impresso. Agora, a quantidade caracteres deve lido pelo teclado e passado à função linha, assim como o caractere a ser impresso.

5) Crie, compile, execute e analise o prog5.c, cujo conteúdo é apresentado abaixo. Finalmente, tente fazer ou responder as seguintes questões:

a) Por que é melhor usar parâmetro em funções do que usar variáveis globais?

b) Tente imprimir o valor de *i* na **main**, sendo que *i* foi declarado dentro da função *imprime\_variavel\_local\_da\_funcao(...)*. Isso é possível, ou seja, a utilização de *i* dessa função no programa principal **main**, após a chamada da função? Justifique.

c) Por que a alteração que *n* sofreu dentro da função *alteracao\_de\_parametro\_dentro\_da\_funcao* não é mantida, após sua chamada, no programa principal **main**? Justifique.

---

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

```
int v; //variável global, pois é externa às funções, inclusive à main
void imprime_variavel_global();
void imprime_variavel_local_da_main(int);
void imprime_variavel_local_da_funcao(int);
void alteracao_de_parametro_dentro_da_funcao(int);
int main(){
    int n; //variável local, pois é interna à main()
    printf("Impressão variável global v passada como argumento.\n");
    printf("Forneça o valor de v (global): ");
    scanf("%d",&v); //leia o valor da variável v global
    imprime_variavel_global();
    printf("-----\n");
    printf("Variável local n da main como argumento.\n");
    printf("Forneça o valor de n (local à main): ");
    scanf("%d",&n); //leia o valor da variável n local
    imprime_variavel_local_da_main(n);
    printf("-----\n");
    imprime_variavel_local_da_funcao(n);
    printf("-----\n");
    alteracao_de_parametro_dentro_da_funcao(n);
    printf("Valor de n na main: %d.\n",n); //alteração de n na função não afetou n de main
}
void imprime_variavel_global(){
    printf("Valor da variável global: %d.\n",v);
}
void imprime_variavel_local_da_main(int n){
    printf("Valor da variável local da main: %d.\n",n);
}
void imprime_variavel_local_da_funcao(int n){
    int i=n; //i é variável local da função
    printf("Valor da variável local da função: %d.\n",i);
}
void alteracao_de_parametro_dentro_da_funcao(int n){
    printf("Valor inicial de n dentro da função: %d.\n",n);
    n = 0;
    printf("Valor final de n dentro da função: %d.\n",n);
}
```