



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO

ATIVIDADE PRÁTICA

KELE PÓLVORA CAIRES – RU: 2707381
SANDRO DE ARAÚJO

IGUAÍ – BA
2019

1 EXERCÍCIO 1

ENUNCIADO: Escreva um algoritmo em linguagem C com as seguintes instruções:

1. Declare três variáveis (inteiro, real e char);
2. Declare três ponteiros;
3. Associe as variáveis aos ponteiros;
4. Modifique os valores de cada variável indiretamente usando os ponteiros associados. Para armazenar os valores nas variáveis, armazene na variável **char** a primeira letra do seu nome, na variável **inteira** os dois últimos dígitos do seu RU e na variável **real** os 4 últimos dígitos do seu RU, sendo os 2 últimos os valores com vírgula;
5. Imprima na tela os valores das variáveis antes e após a modificação.

Solução do aluno:

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

int main() {
    //declarando variáveis
    int x = 10;
    char y = 'A';
    float z = 10.10;

    //declarando ponteiros
    int *a;
    char *b;
    float *c;

    //Associando as variáveis aos ponteiros
    a = &x;
    b = &y;
    c = &z;

    //Imprimindo na tela as variáveis antes de modificar
    printf("\nValores das variaveis antes:%d", x);
    printf("\nValores das variaveis antes:%c", y);
    printf("\nValores das variaveis antes:%.2f", z);

    //Alterando as variáveis utilizando os ponteiros
    *a = 81;
    *b = 'K';
    *c = 73.81;

    //quebra de linha
    printf("\n\n");

    //Imprimindo na tela as variáveis modificada utilizando os ponteiros
    printf("\nApos a modificacao: %d", x);
    printf("\nApos a modificacao:%c", y);
    printf("\nApos a modificacao:%.2f", z);

    //Quebra de linha
    printf("\n\n");

    //finalização do programa
    system("pause");
    return 0;
}
```

```
}
```

Imagem do código funcionando no seu computador:

```
Valores das variaveis antes:10
Valores das variaveis antes:0
Valores das variaveis antes:10.10

Apos a modificacao: 81
Apos a modificacao:K
Apos a modificacao:73.81

Pressione qualquer tecla para continuar. . . _
```

2 EXERCÍCIO 2

ENUNCIADO: Escreva um algoritmo em LINGUAGEM C que armazene na memória o seu RU e o valor 1234567, ambos digitados pelo usuário na tela. Em seguida, imprima na tela ambos RU usando ponteiros. O algoritmo também vai ter que comparar os dois RU usando ponteiros e imprimir na tela qual é o maior.

Solução do aluno:

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

int main(void)
{
    //Variáveis
    int x, y;

    //Ponteiros
    int *a, *b;

    //Associando as variáveis com os ponteiros
    a = &x;
    b = &y;

    //Solicitando na tela a digitação do usuário e guardando na variável a informação digitada
    printf("Digite o primeiro RU:");
    scanf_s("%d", &x);

    printf("Digite o segundo RU:");
    scanf_s("%d", &y);

    //Alterando as variáveis utilizando os ponteiros
    *a = x;
    *b = y;

    //Quebra de linha
    printf("\n\n");
```

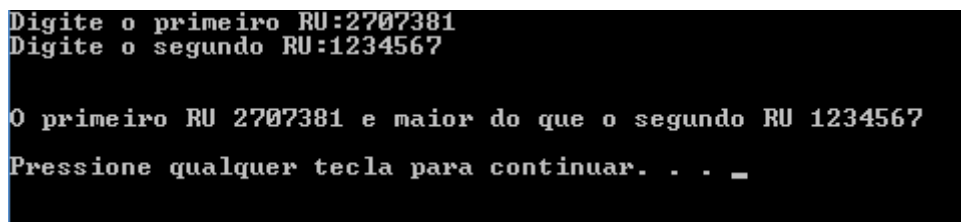
```

//Comparando qual é maior
if (a > b) {
    printf("O primeiro RU %d e maior do que o segundo RU %d", *a, *b);
    printf("\n\n");
}
else
{
    printf("O segundo RU %d e maior do que o primeiro RU %d", *b, *a);
    printf("\n\n");
}

//Finalizando o programa
system("pause");
return 0;
}

```

Imagem do código funcionando no seu computador:



```

Digite o primeiro RU:2707381
Digite o segundo RU:1234567

O primeiro RU 2707381 e maior do que o segundo RU 1234567
Pressione qualquer tecla para continuar. . . _

```

3 EXERCÍCIO 3

ENUNCIADO: Faça um algoritmo em linguagem C com as seguintes funcionalidades:
 Receba um registro, com dois campos, como dados de entrada.

- ☐ O primeiro campo é um vetor que vai armazenar o nome do aluno.
- ☐ O segundo campo é uma variável do tipo inteiro que vai armazenar o RU do aluno.
- ☐ Imprime na tela os dados armazenados na estrutura.

Solução do aluno:

```

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

int main(void)
{
    //Recebendo 2 campos como dados de entrada
    struct aluno
    {
        char nome[5] ="Kele"; // vetor armazenando nome do aluno
        int ru; // variável para armazenar o ru
    };

    struct aluno dados;

    // armazenando o ru do aluno
    dados.ru = 2707381;
}

```

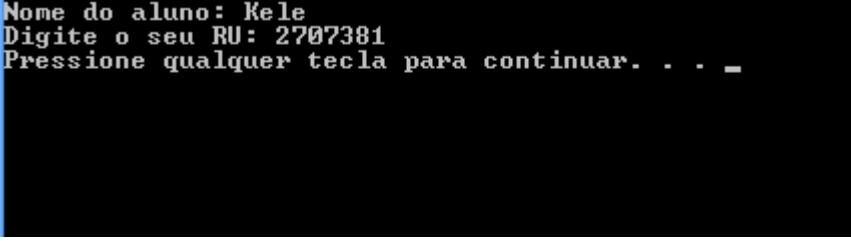
```

        // Mostrando na tela o nome do aluno e do ru
        printf("Nome do aluno: %s\n", dados.nome);
        printf("Digite o seu RU: %d\n", dados.ru);

        //Finalizando o programa
        system("pause");
        return 0;
}

```

Imagem do código funcionando no seu computador:



```

Nome do aluno: Kele
Digite o seu RU: 2707381
Pressione qualquer tecla para continuar. . . _

```

4 EXERCÍCIO 4

ENUNCIADO: Replique o exercício 3. Porém, agora, declare um ponteiro para a estrutura de dados heterogênea. No momento da leitura dos dados e da impressão na tela, use o ponteiro para buscar o conteúdo dos campos. Imprima na tela também o seu RU na tela.

Solução do aluno:

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int main(void)
{
    struct aluno
    {
        char nome[5]; //vetor para armazenar nome do aluno
        int ru; //variável para armazenar o ru do aluno
    };

    struct aluno dados, *p_dados;

    //atribuindo valores
    strcpy_s(dados.nome, "Kele");
    dados.ru = 2707381;

    //Ponteiros
    p_dados = &dados;

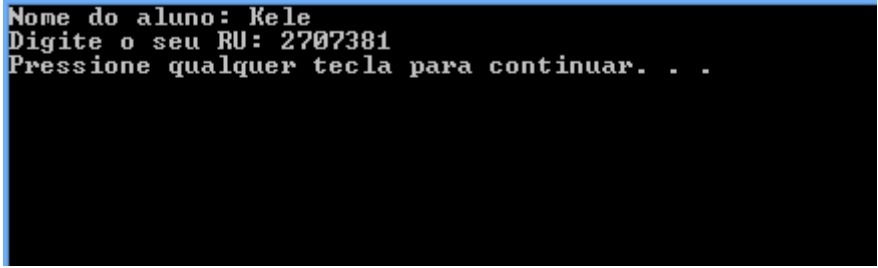
    // Mostrando na tela o nome do aluno e do ru
    printf("Nome do aluno: %s\n", p_dados->nome);
    printf("Digite o seu RU: %d\n", p_dados->ru);

    //Finalizando o programa
    system("pause");
    return 0;
}

```

```
}
```

Imagem do código funcionando no seu computador:



```
Nome do aluno: Kele
Digite o seu RU: 2707381
Pressione qualquer tecla para continuar. . .
```

5 EXERCÍCIO 5

ENUNCIADO Faça um algoritmo em linguagem C que contenha dois números inteiros digitados na tela pelo usuário:

- O primeiro número marca um início;
- O segundo número marca um fim;

O algoritmo vai contar quantos números existem entre o início (primeira entrada) e o fim (segunda entrada). A impressão na tela do usuário deve ser realizada de duas formas:

- Iterativa;
- Recursiva;

Ao colocar no seu relatório uma imagem do seu código funcionando, coloque ele rodando utilizando como valor de início os 2 últimos valores do seu RU e valor final o número 99.

Solução do aluno:

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

int imprimeN(int comeco, int sfim);
int main()
{
    // variáveis
    int comeco, sfim, i;

    printf("Inicio\n");//pedindo o usuário para digitar na tela
    scanf_s("%d", &comeco); // guardando a informação na variável
    printf("Fim\n");
    scanf_s("%d", &sfim);

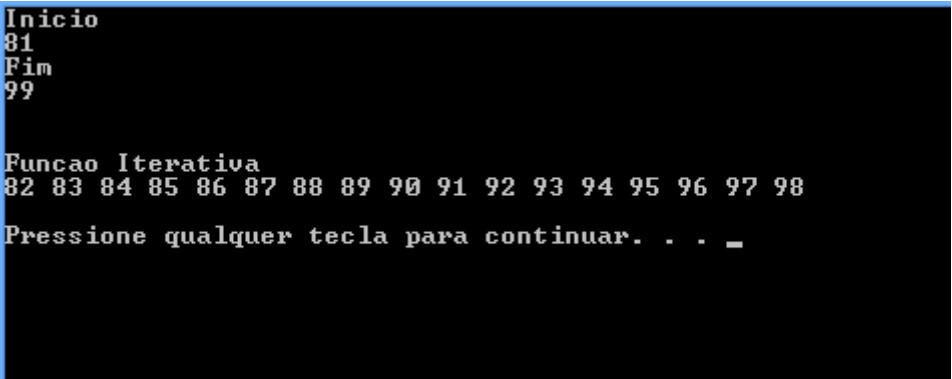
    printf("\n\n"); // quebra de linha

    //mostrando na tela o resultado de forma iterativa
    printf("Números entre os dados digitados a cima\n");
    for (i=comeco+1; i<=sfim-1; i++) {
        printf("%d ", i);
    }
    printf("\n\n");// quebra de linha
```

```
        system("pause");//parando programa
        return 0;
    }

    // função recursiva
    int imprimeN(int comeco, int sfim)
    {
        if (comeco <= sfim) {
            printf_s("%d", comeco);
            imprimeN(comeco +1 , sfim);
        }
    }
}
```

Imagem do código funcionando no seu computador:



```
Inicio
81
Fim
99

Funcao Iterativa
82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98

Pressione qualquer tecla para continuar. . . _
```