



MINISTÉRIO DA EDUCAÇÃO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E  
TECNOLOGIA DO PIAUÍ

CURSO : Análise e Desenvolvimento de Sistemas

### ATIVIDADE

1. Leia dois nomes e compare se os nomes são iguais. Se forem iguais imprima “São Iguais!”, senão imprima “São Diferentes”.

OBSERVAÇÃO:

Faça usando:

```
char nome1[20];
```

```
char nome2[20];
```

USE A FUNÇÃO STRCMP () PARA COMPARAR AS STRINGS.

2. Leia dois nomes e compare se os nomes são iguais. Se forem iguais imprima “São Iguais!”, senão imprima “São Diferentes”.

OBSERVAÇÃO:

Faça usando:

```
char nome1[20];
```

```
char nome2[20];
```

UTILIZE O COMANDO DE REPETIÇÃO “FOR” PARA PERCORRER AS STRINGS CARACTERE A CARACTERE E FAZER A COMPARAÇÃO ENTRE ELAS.

3. Leia dois nomes e compare se os nomes são iguais. Se forem iguais imprima “São Iguais!”, senão imprima “São Diferentes”.

OBSERVAÇÃO: Faça usando:

```
string nome1;
```

```
string nome2;
```

4.

```
#include <stdio.h>
#include <string.h>
#include <iostream>
using namespace std;

typedef struct {
    char valor[10];
} Str;

main() {
    char valor[3]={"um"};

    Str v={"123"};

    int i=0;
    while (i<=strlen(v.valor)){
        cout<<i<<"--"<<v.valor[i]<<endl;
        i++;
    }

    cout<<"-----";

    int j=0;
    while (valor[j]!='\0'){
        cout<<"j ="<<j<<"- "<<valor[j]<<endl;
        j++;
    }

    cout<<j;

}
```

4.1 Considerando o código acima, qual o valor de i e j ao final do programa?

5.

```
#include <stdio.h>
#include <iostream>
using namespace std;

typedef struct no {
    int item;
    struct no *prox;
}No;

main() {

    No v[4]={ {3,v+2}, {1,v+3}, {4,NULL}, {2,v+0}};
```

```

        for (No *p=v+1;p!=NULL;p=p->prox)
            cout<<p->item;

    }

```

6.1 o que será impresso ao final da execucao do programa?

6.2 Inclua os valores no vetor agora através da leitura do teclado.

OBS: LEIA O VALOR DE item através do teclado e inclua no vetor de maneira que o ponteiro prox deve apontar sempre para o proximo elemento do vetor.

5.

```

#include <stdio.h>
#include <iostream>
using namespace std;

typedef struct no {
    int item;
    struct no *prox;
}*Lst;

Lst no(int x, Lst p){
    Lst n = (struct no*)malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

main(){

    Lst p = (no(1,no(2,no(3,no(4,NULL))))) ;

    while (p){
        cout<<p->item<<endl;
        p=p->prox;
    }
}

```

7.1 o que será impresso ao final da execucao do programa?

7.2 Inclua os valores no vetor agora através da leitura do teclado.

OBS: LEIA O VALOR DE item através do teclado e inclua na lista de maneira que

o ponteiro prox deve apontar sempre para o proximo elemento da lista.

8. Sobre o código abaixo:

```
#include <stdio.h>

void x(char *str1, char *str2) {
    while (*str1) {
        str1++;
    }
    while (*str2) {
        *str1 = *str2;
        str1++;
        str2++;
    }
    *str1 = '\0';
}

int main() {
    char str1[100] = "Hello ";
    char str2[] = "World!";
    x(str1, str2);
    printf(" %s\n", str1);
    return 0;
}
```

8.1. O que será impresso ao final do programa?

8.2. O que a função X está fazendo?

9. Sobre o código abaixo:

```
#include <stdio.h>

int x(char *str) {
    int y = 0;
    while (*str) {
        y++;
        str++;
    }
    return y;
}

int main() {
    char str[] = "Hello World";
    int w = x(str);
    printf("%d\n", w);
    return 0;
}
```

9.1. O que será impresso ao final do programa?

9.2. O que a função X está fazendo?

10. Sobre o código abaixo:

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

// Definindo a estrutura do nó
typedef struct Node {
    int data;
    struct Node* next;
} Node;

void append(Node** head_ref, int new_data) {
    Node* new_node = (Node*)malloc(sizeof(Node));
    Node* p = *head_ref;

    new_node->data = new_data;
    new_node->next = NULL;

    if (*head_ref == NULL) {
        *head_ref = new_node;
        return;
    }

    while (p->next != NULL) {
        p = p->next;
    }

    p->next = new_node;
    return;
}

// Função para imprimir a lista
void printList(Node* node) {
    while (node != NULL) {
        printf(" %d ", node->data);
        node = node->next;
    }
}

int main() {
    Node* head = NULL;
    append(&head, 1);
    append(&head, 2);
    append(&head, 3);
    append(&head, 4);
    printf("Lista encadeada: ");
    printList(head);

    return 0;
}
```

10.1. O que será impresso ao final do programa?

10.2. Se o algoritmo está incluindo no final da lista, Faça uma função que inclua no início da lista.

10.3. Se o algoritmo está incluindo no início da lista, Faça uma função que inclua no final da lista.