LISTA PONTEIROS

1 a 5. Fazer upload dos arquivos .c
6. Fazer upload do png

1. Declare uma variável do tipo char chamada **c** e um ponteiro para char chamado **pc**. Inicialize **c** com o valor 'a' e **pc** com o endereço de **c**.

Imprima as seguintes linhas:

a) O endereço de **c** e o valor guardado por ele.

b) O valor de **pc** e o valor guardado no endereço apontado por ele.

c) O endereço de **pc**.

d) O endereço do valor guardado no endereço apontado por **pc** e o valor guardado no endereço de **pc** (**utilize &*p e *&p)** explicando ao mesmo tempo o que é desreferenciação.

e) Explicação do porquê os dois endereços impressos na linha d. são iguais.

f) Dica: Lembre-se de %p para imprimir endereços de memória.

2. Declare um vetor inteiros **int vet[5] = {1,2,3,4,5}** e um ponteiro de inteiros **p** que aponte para esse vetor.

Você deverá imprimir:

A) O endereço guardado em **vet** e o endereço guardado em **p**.
B) Imprimir todos os valores de **vet** utilizando **p** com **[ ]**.
C) Imprimir todos os valores de **vet** utilizando **p** com *.
D) Faça a mesma coisa de **b** e **c** utilizando **vet** no lugar de **p**.

3.

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

int main()
{
 char str[50]; // String
 char str_inv[50]; // String invertida
 char *ptr_str = str;
 char *ptr_inv = str_inv;
 int i=-1;
 scanf(" %s", str);

[...]
```

```
 printf(" O inverso da string : %s\n\n",str_inv);
 return 0;
}
```

A) Usando somente as variáveis **ptr_str**, **ptr_inv e i** e o **while** implemente um código no espaço ***[...]*** que inverta a string str e coloque na **str_inv**. (Não use o operador [] nem str e str_inv e nem funções da string.h. Utilize somente os ponteiros e a aritmética de ponteiros).

B) Explique por quê não é necessário colocar "&" antes de str_inv no "scanf".

4. Faça um programa que aloque dinamicamente um vetor de strings **str**, receba de entrada várias strings enquanto existir entrada no buffer e guarde-as nesse vetor utilizando ponteiros auxiliares.

5. Considere a seguir a seguinte estrutura:

```
struct el_lista{
      int valor;
      struct el_lista *proximo;
};
```

Faça um programa que a cada valor recebido pelo teclado enquanto **valor != 0** aloque uma variável do tipo **el_lista**, guardando o valor e com **proximo** apontado para **NULL** e modifique o anterior alocado apontando **proximo** para o atual. (Utilize aritmética de ponteiros)

6. Ilustre graficamente (desenhe) o que acontece com os ponteiros, memória e blocos de memória quando você realiza um:

A) Aloca um vetor utilizando **malloc**
A) Aloca uma matriz utilizando **malloc**
B) Aloca um vetor utilizando **calloc**
C) Aloca uma matriz utilizando **calloc**

"Não seja um cara triste"
Zyzz