

## PONTEIROS

“Aprendizado vem com a consistência do estudo”

ALESSON SOUSA VIANA

22/09/20

1) Suponha que ted é uma variável tipo double. Declare um ponteiro que aponte para ted e use o ponteiro para mostrar o valor de ted.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
double ted = 10;
double * pot = &ted;
cout << "Valor de ted: " << *pot;
}
```

2) Suponha que moto é um vetor de 10 float. Declare um ponteiro que aponte para o primeiro elemento de moto e use o ponteiro para mostrar o primeiro e o último elemento do vetor.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
float moto[] = {0,1,2,3,4,5,6,7,8,9};
float * pneu = moto;
cout << "Primeiro elemeto: " << *pneu << "\nSegundo elemento: " << *pneu+9;
}
```

3) Escreva um programa que peça ao usuário para digitar um número inteiro positivo e então crie um vetor dinâmico com essa quantidade de inteiros.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
```

```

int * pot;
int aux;
cout << "Digite um numero inteiro: ";
cin >> aux;
pot = new int[aux];
for (int i = 0; i < aux; i++)
{
    cout << "\nVetor " << i << ":" << pot[i] << endl;
}
delete [] pot;
}

```

4) Defina um registro que descreva um peixe. O registro deve incluir o tipo, o peso e o comprimento do peixe. Em seguida mostre como alocar dinamicamente um registro do tipo peixe. Mostre também como ler um valor para cada membro do registro e como imprimir o conteúdo do registro.

```

#include <iostream>
#include <stdio.h>
using namespace std;
struct peixes
{
    char tipo[30];
    int peso;
    float comp;
};
void menu(){
    cout << "\t----Opcao----\n";
    cout << "1 - cadastra\n";
    cout << "2 - Listar\n";
    cout << "3 - Sair\n";
    cout << "=>";
}
int cadastra(peixes peixe[], int qtde)
{

```

```

for (int i = 0; i <= qtde; i++)
{
    cout << "\t----CADASTRA----\n";
    cout << "Tipo: ";
    cin >> peixe[i].tipo;
    cout << "\nPeso: ";
    cin >> peixe[i].peso;
    cout << "\nComprimento: ";
    cin >> peixe[i].comp;
}
system("cls");
}

void Listar(peixes peixe[], int qtde){
    for (int i = 0; i < qtde; i++)
    {
        cout << "\t----LISTAS DE PEIXES----\n";
        cout << "Tipo: " << peixe[i].tipo << "\nPeso: " << peixe[i].peso
            << "\nComprimento: " << peixe[i].comp << endl;
    }
    system("pause");
    system("cls");
}

int main() {
    int tam = 0;
    peixes * peixe;
    peixe = new peixes[tam];
    int opcao;
    int qtdepeixe = 0;
    cout << "Digite tamanho do vetor: ";
    cin >> tam;
    system("cls");
    do

```

```

{
    menu();
    cin >> opcao;
    system("cls");
    switch(opcao)
    {
        case 1: cadastra(peixe, qtdepeixe);
                qtdepeixe++;
                break;
        case 2: Listar(peixe, qtdepeixe);
                break;
    }
} while (opcao!=3);
delete [] peixe;
}

```

5) Escreva o código para criar um vetor dinâmico de peixes. O tamanho do vetor pode ser definido livremente. Mostre como ler e como imprimir o peso do segundo peixe.

```

#include <iostream>
#include <stdlib.h>
using namespace std;
void Menu()
{
    cout << "\t----MENU----\n";
    cout << "[1] - Cadastrar\n";
    cout << "[2] - Listar\n";
    cout << "[3] - Sair\n";
    cout << "=>";
}
void Cadastra(balao *pot, int qtde)
{
    for (int i = 0; i <= qtde; i++)
    {

```

```

        cout << "\t-----CADASTRA-----\n";
        cout << "\nPeso: ";
        cin >> pot[i];
    }
    system("cls");
}

void Listar(int pot[], int qtde)
{
    for (int i = 0; i <= qtde; i++)
    {
        cout << "\t-----LISTAS-----\n";
        cout << "Peso: " << pot[i] << endl;
    }
    system("pause");
    system("cls");
}

int main()
{
    int tam = 0;
    int * pot;
    pot = new int[tam];
    int opcao;
    int qtdepeixe = 0;
    cout << "Digite tamanho do vetor: ";
    cin >> tam;
    system("cls");
    do
    {
        menu();
        cin >> opcao;
        system("cls");
    }

```

```

switch (opcao)
{
case 1:
    cadastra(pot, qtdepeixe);
    qtdepeixe++;
    break;
case 2:
    Listar(pot, qtdepeixe);
    break;
}
} while (opcao != 3);
delete[] pot;
}

```

6) Defina o registro balao como mostrado abaixo. Construa um programa para alocar dinamicamente um registro do tipo balao. Peça ao usuário para entrar com valores para cada um dos membros e em seguida imprima o conteúdo do registro.

```

#include <iostream>
#include <stdlib.h>
using namespace std;
struct balao
{
    char marca[20]; // nome da marca
    int modelo;    // número do modelo
    float diametro; // diâmetro em metros
};
void Menu()
{
    cout << "\t----MENU----\n";
    cout << "[1] - Cadastrar\n";
    cout << "[2] - Listar\n";
    cout << "[3] - Sair\n";
}

```

```

    cout << "=>";
void Cadastra(balao *pot, int qtde)
{
    for (int i = 0; i <= qtde; i++)
    {
        cout << "\t--CADASTRO--\n";
        cout << "Marca: ";
        cin >> pot[i].marca;
        cout << "\nModelo: ";
        cin >> pot[i].modelo;
        cout << "\nDiametro: ";
        cin >> pot[i].diametro;
        system("cls");
    }
}

void Listar(balao *pot, int qtde)
{
    for (int i = 0; i < qtde; i++)
    {
        cout << "Marca: " << pot[i].marca << "\nModelo: " << pot[i].modelo <<
"\nDiametro: " << pot[i].diametro << endl;
    }
    system("pause");
    system("cls");
}

int main()
{
    int tam = 0;
    balao *pot;
    pot = new balao[tam];
    int qtde = 0, op;
    cout << "Digite tamanho do vetor: ";

```

```

cin >> tam;
system("cls");
do
{
    Menu();
    cin >> op;
    switch (op)
    {
        case 1:
            Cadastra(pot, qtde);
            qtde++;
            break;
        case 2:
            Listar(pot, qtde);
            break;
    }
} while (op != 3);
delete [] pot;
}

```

7) Construa um registro para guardar informações sobre um carro. Um carro deve ter nome do modelo, ano de fabricação e preço. Em seguida construa um vetor de 10 carros inicializando os dois primeiros carros para {"Vectra", 2009, R\$58.000,00} e {"Polo", 2008, R\$45.000,00}, respectivamente. Use um ponteiro para apontar para o segundo carro e imprimir seus dados.

```

#include <iostream>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
struct carro
{
    char modelo[30];
    int ano;

```



```

        float preco;
    };

    int main()
    {
        carro carros[10] = {
            {"polo", 2008, 58.000},
            {"fiat", 2020, 90.000}};

        carro * pot = carros;

        cout << "Modelo: " << pot[1].modelo << "\nAno: " << pot[1].ano << "\nPreco: " << pot[1].preco << " Mil" << endl;
    }

```

8) Repita o exercício anterior criando um vetor dinâmico de carros. Peça ao usuário para digitar os dados de dois carros e em seguida imprima o valor total dos carros.

```

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

struct carro
{
    char modelo[30];
    int ano;
    float preco;
};

int cadastra(carro carros[])
{
    cout << "\t-----CADASTRA CARRO 1-----\n";
    cout << "Modelo: ";
    cin >> carros[0].modelo;
    cout << "\nAno: ";
    cin >> carros[0].ano;
}

```

```

        cout << "\nPreço: ";
        cin >> carros[0].preco;

        cout << "\n\t-----CADASTRA CARRO 1-----\n";

        cout << "Modelo: ";
        cin >> carros[1].modelo;

        cout << "\nAno: ";
        cin >> carros[1].ano;

        cout << "\nPreço: ";
        cin >> carros[1].preco;
    }

    void Listar(carro carros[])
    {
        float total = carros[0].preco + carros[1].preco;

        cout << "\t-----LISTAS-----\n";

        cout << "Valor total: " << total;
    }

    int main()
    {
        int tam = 10;

        carro * carros;

        carros = new carro[tam];

        int opcao;

        int qtdecarros = 0;

        cadastra(carros);

        Listar(carros);

        delete[] carros;
    }

```

