

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
//using namespace std;
//
//int main() {
//
//
//    system("pause");
//    return 0;
//}

```

```

=====
=====

```

```

/*
1) Faça um programa que declare uma variável X do tipo inteira e atribua a ela o valor 100.
Em seguida crie e inicialize o ponteiro ptr com a variável X.
Exiba o conteúdo do ponteiro.
*/

```

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
//using namespace std;
//
//int main() {
//    int x;
//    int *ptr;
//
//    x = 100;
//    ptr = &x;
//
//    cout << "O valor de ptr: " << *ptr << endl;
//
//    system("pause");
//    return 0;
//}

```

```

/*
2) Faça um programa que declare um ponteiro ptr do tipo inteiro,
aloque memória para ele, atribua o valor 500 e exiba seu valor.
*/

```

```

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

```

```
//      int *ptr;
//
//      ptr = new int;
//      *ptr = 500;
//
//      cout << "O valor de ptr: " << *ptr << endl;
//
//      delete ptr;
//
//      system("pause");
//      return 0;
//}
```

/\*

3) Faça um programa que declare duas variáveis X e Y do tipo inteira. Atribua a X o valor 100 e a Y o valor 400. Em seguida crie e inicialize dois ponteiros ptrX e ptrY com a variável X e Y. Some os ponteiros e exiba o resultado.

\*/

```
//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//int main() {
//      int x, y, total;
//      int *ptrX, *ptrY;
//
//      x = 100;
//      y = 400;
//
//      ptrX = &x;
//      ptrY = &y;
//
//      total = *ptrX + *ptrY;
//
//      cout << "Total: " << total << endl;
//
//      system("pause");
//      return 0;
//}
//
```

/\*

4) Crie uma variável X do tipo inteiro e passa-a para a função atribuirValor(int \*ptr). Esta função deve atribuir o valor 100 ao ponteiro e retornar para a função main. Em seguida, na função main, imprimir o valor.

\*/

```
//#include <iostream>
```

```

#include <stdlib.h>
#include <string>
//
using namespace std;
//
void atribuirValor(int *ptr);
//
int main() {
    int x;
//
    atribuirValor(&x);
//
    cout << "X: " << x << endl;
//
    system("pause");
    return 0;
//}
//
void atribuirValor(int *ptr) {
    *ptr = 100;
//}

```

/\*  
5) Crie 3 variáveis do tipo inteiro e atribua os valores: 100, 200, 300.  
Em seguida crie uma função que tenha como parâmetro 3 ponteiros do tipo inteiro.  
Passe as 3 variáveis para a função.  
A função deve somar os 3 valores e retornar o resultado para a função main.  
Na função main, exiba o resultado da soma.  
\*/

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
using namespace std;
//
int somarValor(int *ptrX, int *ptrY, int *ptrZ);
//
int main() {
    int x, y, z, total;
//
    x = 100;
    y = 200;
    z = 300;
//
    total = somarValor(&x, &y, &z);
//
    cout << "Total: " << total << endl;
//
    system("pause");
    return 0;
//}
//
int somarValor(int *ptrX, int *ptrY, int *ptrZ) {

```

```
//      int total;
//
//      total = *ptrX + *ptrY + *ptrZ;
//
//      return total;
//}
```

```
/*include <iostream>
#include <stdlib.h>
#include <string>
```

```
using namespace std;
```

```
void calcularValor(int *ptrX, int *ptrY, int *ptrZ, int *resSoma, int *resSub);
```

```
int main() {
    int x, y, z, totalSoma, totalSub;

    x = 100;
    y = 200;
    z = 300;

    calcularValor(x, y, z, &totalSoma, &totalSub);

    cout << "Soma: " << totalSoma << endl;
    cout << "Subtração: " << totalSub << endl;

    system("pause");
    return 0;
}
```

```
void calcularValor(int ptrX, int ptrY, int ptrZ, int *resSoma, int *resSub) {

    *resSoma = ptrX + ptrY + ptrZ;
    *resSub = ptrX - ptrY - ptrZ;

}*/
```

```
/*
6) Faça um programa que declare uma variável X do tipo inteira e atribua
a ela o valor 100.
Em seguida crie e inicialize o ponteiro ptr com a variável X.
Exiba o endereço do ponteiro.
*/
//
//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
```

```
//
//int main() {
//    int x;
//    int *ptr;
//
//    x = 100;
//    ptr = &x;
//
//    cout << "O Endereço de X: " << &x << endl;
//    cout << "O Endereço de ptr: " << ptr << endl;
//    cout << "O Endereço de ptr: " << &ptr << endl << endl;
//
//
//    system("pause");
//    return 0;
//}
```

/\*  
7) Faça um programa que declare um vetor do tipo inteiro de 5 posições e atribua valores a ele.  
Em seguida crie e inicialize o ponteiro ptr com o vetor.  
Exiba os endereços das 5 posições do ponteiro (formato vetor).  
Exiba os valores das 5 posições do ponteiro (formato vetor).  
Exiba os endereços das 5 posições do ponteiro (formato ponteiro).  
Exiba os valores das 5 posições do ponteiro (formato ponteiro).  
\*/

```
//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//int main() {
//    int x[5] = {10, 20, 30, 40, 50};
//    int *ptr;
//
//    ptr = x;
//
//    // Formato VETOR
//    cout << "O Endereço da 1 posição é: " << &ptr[0] << endl;
//    cout << "O Endereço da 3 posição é: " << &ptr[2] << endl;
//    cout << "O Endereço da 5 posição é: " << &ptr[4] << endl;
//    cout << "O Valor da 2 posição é: " << ptr[1] << endl;
//    cout << "O Valor da 4 posição é: " << ptr[3] << endl;
//
//    cout << endl << endl;
//
//    // Formato PONTEIRO
//    cout << "O Endereço da 1 posição é: " << (ptr + 0) << endl;
//    cout << "O Endereço da 3 posição é: " << (ptr + 2) << endl;
//    cout << "O Endereço da 5 posição é: " << (ptr + 4) << endl;
//    cout << "O Valor da 2 posição é: " << *ptr + 1 << endl;
//    cout << "O Valor da 4 posição é: " << *ptr + 3 << endl;
```

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

/\*  
8) Faça um programa que alogue um ponteiro do tipo inteiro com 5 posições de memória.

Utilizando notação de ponteiro faça:

- 1) Atribua valores com cin.
- 2) Exiba o endereço da primeira, terceira e quinta posição do ponteiro.
- 3) Exiba os valores da segunda e quarta posição do ponteiro.

```
*/

//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//int main() {
//    int *ptr;
//    int k;
//
//    ptr = new int[5];
//
//    for (k = 0; k < 5; k++) {
//        cout << "informe o valor para a " << (k + 1) << " posição: ";
//        cin >> *(ptr + k);
//    }
//
//    cout << "O Endereço da 1 posição é: " << (ptr + 0) << endl;
//    cout << "O Endereço da 3 posição é: " << (ptr + 2) << endl;
//    cout << "O Endereço da 5 posição é: " << (ptr + 4) << endl;
//    cout << "O Valor da 2 posição é: " << *(ptr + 1) << endl;
//    cout << "O Valor da 4 posição é: " << *(ptr + 3) << endl;
//
//    delete [] ptr;
//
//    system("pause");
//    return 0;
//}
```

/\*  
9) Faça um programa que alogue um ponteiro do tipo inteiro com 5 posições de memória.  
Atribua valores com cin (utilize notação de ponteiro).  
Crie um vetor de 5 posições do tipo inteiro.  
Copie os valores do ponteiro (utilize notação de ponteiro) para o vetor.  
Exiba os valores do vetor.

```
*/

//#include <iostream>
//#include <stdlib.h>
```

```

//#include <string>
//
//using namespace std;
//
//int main() {
//    int *ptr;
//    int vetor[5];
//    int k;
//
//    ptr = new int[5];
//
//    // Entrada dos dados
//    for (k = 0; k < 5; k++) {
//        cout << "informe o valor para a " << (k + 1) << " posição: ";
//        cin >> *(ptr + k);
//    }
//
//    // Cópia dos valores do ponteiro para o vetor
//    for (k = 0; k < 5; k++) {
//        vetor[k] = *(ptr + k);
//    }
//
//    for (k = 0; k < 5; k++) {
//        cout << "O valor para a " << (k + 1) << " posição é: " << vetor[k] << endl;
//    }
//
//    delete[] ptr;
//
//    system("pause");
//    return 0;
//}

```

/\*

10) Faça um programa que declare um vetor X do tipo inteiro de 5 posições e atribua valores a ele.  
 Crie um segundo vetor Y de 5 posições mas deixe-o vazio.  
 Crie uma função que copie os valores de X para Y.  
 A função deve receber os vetores com parâmetros do tipo ponteiro.  
 A cópia deve ser feita no formato de ponteiro.  
 Exiba os valores no vetor y na função main.

\*/

```

//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//void copiaVetores(int x[], int *ptrY);
//
//int main() {
//    int x[5] = { 10, 20, 30, 40, 50 };
//    int y[5];
//    int k;
//

```

```

//      copiaVetores(x, y);
//
//      for (k = 0; k < 5; k++) {
//          cout << "O valor para a " << (k + 1) << " posição é: " << y[k] << endl;
//      }
//
//      system("pause");
//      return 0;
//}
//
//void copiaVetores(int x[], int *ptrY) {
//    int k;
//
//    // Cópia dos valores do ponteiro para o vetor
//    for (k = 0; k < 5; k++) {
//        *(ptrY + k) = x[k];
//    }
//}

```

/\*  
11) Faça um programa que declare 3 vetores X, Y e Z do tipo inteiro de 5 posições.  
Atribua valores para os vetores X e Y;  
Crie uma função que some os valores de X e Y de cada posição e atribua  
resultado na mesma posição no vetor Z.  
Utiliza ponteiros nos parâmetros da função.  
Utilize notação de ponteiros para fazer a soma e atribuição de valores.  
Após o retorno da função exiba os valores do vetor Z.  
\*/

```

//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//void somarVetores(int *ptrX, int *ptrY, int *ptrZ);
//
//int main() {
//    int x[5] = { 10, 20, 30, 40, 50 };
//    int y[5] = { 10, 20, 30, 40, 50 };
//    int z[5];
//    int k;
//
//    somarVetores(x, y, z);
//
//    for (k = 0; k < 5; k++) {
//        cout << "O valor para a " << (k + 1) << " posição é: " << z[k] << endl;
//    }
//
//    system("pause");
//    return 0;
//}
//
//void somarVetores(int *ptrX, int *ptrY, int *ptrZ) {
//    int k;

```



```
//
//      // Cópia dos valores do ponteiro para o vetor
//      for (k = 0; k < 5; k++) {
//          *(ptrZ + k) = *(ptrX + k) + *(ptrY + k);
//      }
//}
```

```
/*=====
=====
```

Para os próximos exercícios deve se utilizar a seguinte estrutura chamada "Veiculo" para armazenar os seguintes dados:

```
.: Modelo do veículo
.: Ano de fabricação
.: Quilometragem
.: Valor de venda
```

```
=====
=====*/
```

```
/*
12) Crie um programa que preencha e exiba os valores da estrutura do Veículo
*/
```

```
//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//struct Veiculo
//{
//    string modelo;
//    int ano;
//    int km;
//    float valor;
//};
//
//int main() {
//    Veiculo carro;
//
//    carro.modelo = "Gol";
//    carro.ano = 2015;
//    carro.km = 30000;
//    carro.valor = 25000.00f;
//
//    cout << "===== " << endl;
//    cout << "Informações sobre o veículo" << endl;
//    cout << "-----" << endl;
//    cout << "Modelo: " << carro.modelo << endl;
//    cout << "Anbo: " << carro.ano << endl;
//    cout << "Km: " << carro.km << endl;
//    cout << "Valor: " << carro.valor << endl;
//    cout << "===== " << endl << endl;
//
//    system("pause");
//    return 0;
```

```
//}
```

```
/*
```

13) Crie um programa que preencha os valores da estrutura do Veículo.  
Em seguida atribua a estrutura para um ponteiro.  
Exibe os dados do veículo utilizando notação de ponteiro.

```
*/
```

```
//#include <iostream>
```

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

```
//#include <string>
```

```
//
```

```
//using namespace std;
```

```
//
```

```
//struct Veiculo
```

```
//{
```

```
//    string modelo;
```

```
//    int ano;
```

```
//    int km;
```

```
//    float valor;
```

```
//};
```

```
//
```

```
//int main() {
```

```
//    Veiculo carro;
```

```
//    Veiculo *ptrCar;
```

```
//
```

```
//    carro.modelo = "Gol";
```

```
//    carro.ano = 2015;
```

```
//    carro.km = 30000;
```

```
//    carro.valor = 25000.00;
```

```
//
```

```
//    ptrCar = &carro;
```

```
//
```

```
//    cout << "=====" << endl;
```

```
//    cout << "Informações sobre o veículo" << endl;
```

```
//    cout << "-----" << endl;
```

```
//    cout << "Modelo: " << ptrCar->modelo << endl;
```

```
//    cout << "Ano: " << ptrCar->ano << endl;
```

```
//    cout << "Km: " << ptrCar->km << endl;
```

```
//    cout << "Valor: " << ptrCar->valor << endl;
```

```
//    cout << "=====" << endl << endl;
```

```
//
```

```
//    system("pause");
```

```
//    return 0;
```

```
//}
```

```
/*
```

14) Crie um programa que preencha os valores da estrutura do Veículo.

Em seguida crie uma função que recebe a estrutura no formato de ponteiro.

Exibe os dados do veículo utilizando notação de ponteiro.

```
*/
```

```
//#include <iostream>
```

```

#include <stdlib.h>
#include <string>
//
using namespace std;
//
struct Veiculo
//{
//    string modelo;
//    int ano;
//    int km;
//    float valor;
//};
//
//void exibirDados(Veiculo *ptrCar);
//
//int main() {
//    Veiculo carro;
//
//    carro.modelo = "Gol";
//    carro.ano = 2015;
//    carro.km = 30000;
//    carro.valor = 25000.00;
//
//    exibirDados(&carro);
//
//    system("pause");
//    return 0;
//}
//
//void exibirDados(Veiculo *ptrCar) {
//    cout << "=====" << endl;
//    cout << "Informações sobre o veículo" << endl;
//    cout << "-----" << endl;
//    cout << "Modelo: " << ptrCar->modelo << endl;
//    cout << "Anbo: " << ptrCar->ano << endl;
//    cout << "Km: " << ptrCar->km << endl;
//    cout << "Valor: " << ptrCar->valor << endl;
//    cout << "=====" << endl << endl;
//}

```

/\*

15) Crie um programa que preencha um vetor de 3 posições com a estrutura Veículo. Em seguida exiba os valores.

\*/

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
using namespace std;
//
struct Veiculo
//{
//    string modelo;

```

```

//      int ano;
//      int km;
//      float valor;
//};
//
//int main() {
//      Veiculo carro[3];
//      int k;
//
//      carro[0].modelo = "Gol";
//      carro[0].ano = 2015;
//      carro[0].km = 30000;
//      carro[0].valor = 25000.00;
//
//      carro[1].modelo = "Mobili";
//      carro[1].ano = 2017;
//      carro[1].km = 5000;
//      carro[1].valor = 30000.00;
//
//      carro[2].modelo = "Civic";
//      carro[2].ano = 2019;
//      carro[2].km = 0;
//      carro[2].valor = 100000.00;
//
//      for (k = 0; k < 3; k++) {
//              cout << "===== " << endl;
//              cout << "Informações sobre o veículo" << endl;
//              cout << "-----" << endl;
//              cout << "Modelo: " << carro[k].modelo << endl;
//              cout << "Anbo: " << carro[k].ano << endl;
//              cout << "Km: " << carro[k].km << endl;
//              cout << "Valor: " << carro[k].valor << endl;
//              cout << "===== " << endl << endl;
//      }
//
//      system("pause");
//      return 0;
//}

```

/\*

16) Crie um programa que preencha um vetor de 3 posições com a estrutura Veículo. Atribua esse vetor a um ponteiro e exiba os dados utilizando notação de ponteiro.

\*/

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
//using namespace std;
//
//struct Veiculo
//{
//      string modelo;
//      int ano;

```

```

//      int km;
//      float valor;
//};
//
//int main() {
//      Veiculo carro[3];
//      Veiculo *ptrCar;
//      int k;
//
//      carro[0].modelo = "Gol";
//      carro[0].ano = 2015;
//      carro[0].km = 30000;
//      carro[0].valor = 25000.00;
//
//      carro[1].modelo = "Mobili";
//      carro[1].ano = 2017;
//      carro[1].km = 5000;
//      carro[1].valor = 30000.00;
//
//      carro[2].modelo = "Civic";
//      carro[2].ano = 2019;
//      carro[2].km = 0;
//      carro[2].valor = 100000.00;
//
//      ptrCar = carro;
//
//      for (k = 0; k < 3; k++) {
//              cout << "===== " << endl;
//              cout << "Informações sobre o veículo" << endl;
//              cout << "-----" << endl;
//              cout << "Modelo: " << (ptrCar+k)->modelo << endl;
//              cout << "Ano: " << (ptrCar + k)->ano << endl;
//              cout << "Km: " << (ptrCar + k)->km << endl;
//              cout << "Valor: " << (ptrCar + k)->valor << endl;
//              cout << "===== " << endl << endl;
//      }
//
//      system("pause");
//      return 0;
//}

```

/\*

17) Crie um programa que preencha um vetor de 3 posições com a estrutura Veículo. Crie uma função que receba o vetor (em formato de ponteiro) e exiba as informações. Utilize notação de ponteiro.

\*/

```

//#include <iostream>
//#include <stdlib.h>
//#include <string>
//
//using namespace std;
//
//struct Veiculo

```

```

//{
//    string modelo;
//    int ano;
//    int km;
//    float valor;
//};
//
//void exibirDados(Veiculo *ptrCar);
//
//int main() {
//    Veiculo carro[3];
//
//    carro[0].modelo = "Gol";
//    carro[0].ano = 2015;
//    carro[0].km = 30000;
//    carro[0].valor = 25000.00;
//
//    carro[1].modelo = "Mobili";
//    carro[1].ano = 2017;
//    carro[1].km = 5000;
//    carro[1].valor = 30000.00;
//
//    carro[2].modelo = "Civic";
//    carro[2].ano = 2019;
//    carro[2].km = 0;
//    carro[2].valor = 100000.00;
//
//    exibirDados(carro);
//
//    system("pause");
//    return 0;
//}
//
//void exibirDados(Veiculo *ptrCar) {
//    int k;
//
//    for (k = 0; k < 3; k++) {
//        cout << "=====" << endl;
//        cout << "Informações sobre o veículo" << endl;
//        cout << "-----" << endl;
//        cout << "Modelo: " << (ptrCar + k)->modelo << endl;
//        cout << "Anbo: " << (ptrCar + k)->ano << endl;
//        cout << "Km: " << (ptrCar + k)->km << endl;
//        cout << "Valor: " << (ptrCar + k)->valor << endl;
//        cout << "=====" << endl << endl;
//    }
//}

```

/\*

18) Crie um programa com dois vetores (carroX e carroY) de 3 posições da estrutura Veículo.

Preencha o primeiro vetor.

Crie uma função que copie o vetor carroX para o vetor carroY.

Utilize ponteiro como parâmetro da função.

A cópia de ser utilizada com notação de ponteiro.  
Após a função exiba os dados do carroY.

```
*/

#include <iostream>
#include <stdlib.h>
#include <string>
//
//using namespace std;
//
//struct Veiculo
//{
//    string modelo;
//    int ano;
//    int km;
//    float valor;
//};
//
//void copiarDados(Veiculo *ptrCarX, Veiculo *ptrCarY);
//
//int main() {
//    Veiculo carroX[3];
//    Veiculo carroY[3];
//    int k;
//
//    carroX[0].modelo = "Gol";
//    carroX[0].ano = 2015;
//    carroX[0].km = 30000;
//    carroX[0].valor = 25000.00;
//
//    carroX[1].modelo = "Mobili";
//    carroX[1].ano = 2017;
//    carroX[1].km = 5000;
//    carroX[1].valor = 30000.00;
//
//    carroX[2].modelo = "Civic";
//    carroX[2].ano = 2019;
//    carroX[2].km = 0;
//    carroX[2].valor = 100000.00;
//
//    copiarDados(carroX, carroY);
//
//    for (k = 0; k < 3; k++) {
//        cout << "=====" << endl;
//        cout << "Informações sobre o veículo" << endl;
//        cout << "-----" << endl;
//        cout << "Modelo: " << carroY[k].modelo << endl;
//        cout << "Anbo: " << carroY[k].ano << endl;
//        cout << "Km: " << carroY[k].km << endl;
//        cout << "Valor: " << carroY[k].valor << endl;
//        cout << "=====" << endl << endl;
//    }
//
//    system("pause");
//    return 0;
```

```

//}
//
//void copiarDados(Veiculo *ptrCarX, Veiculo *ptrCarY) {
//    int k;
//
//    for (k = 0; k < 3; k++) {
//        (ptrCarY + k)->modelo = (ptrCarX + k)->modelo;
//        (ptrCarY + k)->ano = (ptrCarX + k)->ano;
//        (ptrCarY + k)->km = (ptrCarX + k)->km;
//        (ptrCarY + k)->valor = (ptrCarX + k)->valor;
//    }
//}

```

```

/*
19) Aloque memória para uma estrutura do Veículo.
Em seguida preencha e exiba os dados da estrutura.
*/

```

```

#include <iostream>
#include <stdlib.h>
#include <string>
//
using namespace std;
//
struct Veiculo
//{
//    string modelo;
//    int ano;
//    int km;
//    float valor;
//};
//
//int main() {
//    Veiculo *ptrcar;
//
//    ptrcar = new Veiculo;
//
//    ptrcar->modelo = "Gol";
//    ptrcar->ano = 2015;
//    ptrcar->km = 30000;
//    ptrcar->valor = 25000.00;
//
//    cout << "=====" << endl;
//    cout << "Informações sobre o veículo" << endl;
//    cout << "-----" << endl;
//    cout << "Modelo: " << ptrcar->modelo << endl;
//    cout << "Anbo: " << ptrcar->ano << endl;
//    cout << "Km: " << ptrcar->km << endl;
//    cout << "Valor: " << ptrcar->valor << endl;
//    cout << "=====" << endl << endl;
//
//    delete ptrcar;
//
//    system("pause");

```



```
//      return 0;
//}
```

```
/*
```

20) Crie um programa que preencha um ponteiro de 3 posições com a estrutura Veículo.

Em seguida exiba os valores. Utilize notação de ponteiro.

```
*/
```

```
#include <iostream>
#include <stdlib.h>
#include <string>
```

```
using namespace std;
```

```
struct Veiculo
{
    string modelo;
    int ano;
    int km;
    float valor;
};
```

```
int main() {
    Veiculo *ptrCar;
    int k;

    ptrCar = new Veiculo[3];

    (ptrCar + 0)->modelo = "Gol";
    (ptrCar + 0)->ano = 2015;
    (ptrCar + 0)->km = 30000;
    (ptrCar + 0)->valor = 25000.00;

    (ptrCar + 1)->modelo = "Mobili";
    (ptrCar + 1)->ano = 2017;
    (ptrCar + 1)->km = 5000;
    (ptrCar + 1)->valor = 30000.00;

    (ptrCar + 2)->modelo = "Civic";
    (ptrCar + 2)->ano = 2019;
    (ptrCar + 2)->km = 0;
    (ptrCar + 2)->valor = 100000.00;

    for (k = 0; k < 3; k++) {
        cout << "=====" << endl;
        cout << "Informações sobre o veículo" << endl;
        cout << "-----" << endl;
        cout << "Modelo: " << (ptrCar + k)->modelo << endl;
        cout << "Anbo: " << (ptrCar + k)->ano << endl;
        cout << "Km: " << (ptrCar + k)->km << endl;
        cout << "Valor: " << (ptrCar + k)->valor << endl;
        cout << "=====" << endl << endl;
    }
}
```

```
delete [] ptrCar;
```

```
system("pause");
```

```
return 0;
```

```
}
```