1. Execute o código abaic e análise os seguintes resultados:

• Os valores impressos na linha 2 e 5 foram iguais? Justifique

• Os valores impressos na linha 3 e 6 foram iguais? Justifique

|  |
| --- |
| #include<stdio.h>  // A função imprimeEnderecos imprime vários endereços de memória.  //Cada endereço representa o "local" na memória que uma determinada variável foi alocada  // %p é o formato para impressão de endereços de memória  // em C, o vetor v é um ponteiro estático que guarda o endereço do primeiro elemento do vetor  // ponteiro estático é assim chamado porque o endereço apontado por ele nunca muda, ou seja,  //um ponteiro estático aponta sempre para o mesmo endereço  // Logo v vai guardar sempre o mesmo endereço durante a execução da função imprimeEnderecos  // Já vimos o conceito de ponteiro e já trabalhamos com ele. Mas diferente de um vetor, uma variável declarada como //ponteiro representa um ponteiro dinâmico que pode apontar para vários "locais" (endereços)  //diferentes da memória durante a execução de uma mesma função. A única condição é que o endereço apontado pelo //ponteiro seja um endereço válido para o programa  void imprimeEnderecos(int v[2]){  // Antes de cada comando da função, o comentário descreve o que será impresso  // o endereço apontado por v = o endereço armazenado em v  printf("\nO endereço apontado por v na funcaoTeste = %p", v); //5  // o endereço do primeiro elemento do vetor v  printf("\nO endereço do primeiro elemento de v na funcaoTeste = %p", &v[0]); //6  }  int main (){  int v[] = {10,20};  // Antes de cada comando, o comentário descreve o que será impresso  // o endereço apontado por v = o endereço armazenado em v  printf("\nO endereço apontado por v na main = %p", v); // 2  // o endereço apontado por v = o endereço armazenado em v  printf("\nO endereço do primeiro elemento de v na main = %p", &v[0]); //3  imprimeEnderecos(v);  system("pause");  return 0;  }  2. Qual é o resultado do seguinte programa?  #include <stdio.h>  void main(){  float vet[5] = {1.1,2.2,3.3,4.4,5.5};  float \*f;  int i;  f = vet;  printf("contador/valor/valor/endereco/endereco");  for(i = 0 ; i <= 4 ; i++){  printf("\ni = %d",i);  printf(" vet[%d] = %f",i, vet[i]);  printf(" \*(f + %d) = %f",i, \*(f+i));  printf(" &vet[%d] = %p",i, &vet[i]);  printf(" (f + %d) = %p",i, f+i);  }  }  3. Assumindo que pulo[10] é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento?  a) \*(pulo + 2) b) \*(pulo + 4) c) pulo + 4 d) pulo + 2  4. Supor a declaração: int mat[4], \*p, x; Quais expressões são válidas? Justifique:  a) p = mat + 1; b) p = mat++; c) p = ++mat; d) x = (\*mat)++;  5. O que fazem os seguintes programas?  #include <stdio.h>  void main(){  int vet[] = {4,9,13};  int i;  for(i=0;i<3;i++){  printf("%d ",\*(vet+i));  }  }  #include <stdio.h>  void main(){  int vet[] = {4,9,13};  int i;  for(i=0;i<3;i++){  printf("%p ",vet+i);  }  } |

RESPOSTAS

* **Sim, pois o parâmetro enviado da main para a função teste é o mesmo e não foi alterado o valor em nenhuma das funções antes de exibir o resultado.**
* **Sim foram iguais, pois as duas funções indicam a mesma posição do Vetor v[0]. Aliás v[0] = v; pois o endereço da primeira posição do vetor é igual ao endereço do próprio vetor.**

1. **contador/valor/valor/endereco/endereco**

**i = 0 vet[0] = 1.100000 \*(f + 0) = 1.100000 &vet[0] = 0060FEF4 (f + 0) = 0060FEF4**

**i = 1 vet[1] = 2.200000 \*(f + 1) = 2.200000 &vet[1] = 0060FEF8 (f + 1) = 0060FEF8**

**i = 2 vet[2] = 3.300000 \*(f + 2) = 3.300000 &vet[2] = 0060FEFC (f + 2) = 0060FEFC**

**i = 3 vet[3] = 4.400000 \*(f + 3) = 4.400000 &vet[3] = 0060FF00 (f + 3) = 0060FF00**

**i = 4 vet[4] = 5.500000 \*(f + 4) = 5.500000 &vet[4] = 0060FF04 (f + 4) = 0060FF04**

1. **a – \*(pulo + 2)**
2. **As expressões da questão A e D conforme o txt no drive. A questão A irá mostrar a 2ª posição do vetor e a questão D exibirá o primeiro elemento do vetor visto que x recebe o endereço do vetor que é igual ao endereço da primeira posição do Vetor.**
3. **O primeiro programa está imprimindo os valores do vetor visto que vet[i] == \*(vet+i).**

**O segundo programa está imprimindo os endereços das posições do vetor.**