**PONTEIROS E ALOCAÇÃO DINÂMICA DE MEMÓRIA**

1. O que é um ponteiro? Para que serve um ponteiro?
2. Explique as declarações abaixo:
   1. int \*x;
   2. int x;
3. Quais as maneiras de atribuir valor à variável a, após as declarações abaixo:
   1. int a, \*p;
4. Na expressão **“float \*pont;”**, o que é do tipo float?
   1. A variável pont;
   2. O endereço de pont;
   3. A variável apontada por pont;
   4. Nenhuma das anteriores.
5. Considerando que o endereço da variável inteira x foi atribuído a um ponteiro p (1), quais das seguintes expressões estão corretas? Justifique:
   1. x == &p;
   2. x == \*p;
   3. p == \*x;
   4. p == &x;
6. Considere o seguinte trecho de código e responda:

int x;

int \*p;

p = &x;

qual dos comandos abaixo estão corretos (justifique sua resposta):

* 1. scanf(“%i”, &x);
  2. scanf(“%i”,\*x);
  3. scanf(“%i”,p);
  4. scanf(“%i”,&p);

1. Considerando que o endereço da variável x foi atribuída a um ponteiro p, escreva as instruções necessárias para dividir o valor de x por 5, sem utilizar diretamente a variável x.
2. Considerando o código abaixo, quais valores serão impressos?

int main(){

int i=3, j=5;

int \*p = &i, \*q = &j;

printf("%i\n",p == &i);

printf("%i\n",\*p-\*q);

printf("%i\n",\*\*&p);

return 1;

}

1. Qual a saída deste programa?

int main(){

int i=5, \*p=&i;

printf(“%p %d %d %d %d \n”, p, \*p+2, \*\*&p, 3\*\*p, \*\*&p+4);

}

1. Considerando as declarações do exercício 8 quais das seguintes atribuições estão incorretas? Justifique:
   1. p=&i;
   2. \*q=&j;
   3. p=&\*&i;
   4. i=(\*&)j;.
   5. i=\*&\*&j;
   6. q=&p;
   7. i=(\*p)+++\*q;
2. O seguinte programa está correto? Justifique sua resposta:

int main(){

int \*pq;

pq = (int \*) malloc(sizeof(int));

\*pq = 3;

printf("Resultado: %d\n",++\*pq);

return 1;

}

1. Considere o seguinte trecho de programa:

int i=3,j=5;

int \*p, \*q;

p = &i;

q = &j;

Qual é o valor das seguintes expressões?

* 1. p == &i;
  2. \*p - \*q;
  3. \*\*&p;
  4. 3\* -\*p/(\*q)+7;

1. Quais serão as saídas do seguinte programa?

#include <stdio.h>

int main() {

int valor;

int \*p1;

float temp;

float \*p2;

char aux;

char \*nome = "Algoritmos";

char \*p3;

int idade;

int vetor[3];

int \*p4;

int \*p5;

/\* (a) \*/

valor = 10;

p1 = &valor;

\*p1 = 20;

printf("(a) %d \n", valor);

/\* (b) \*/

temp = 26.5;

p2 = &temp;

\*p2 = 29.0;

printf("(b) %.1f \n", temp);

/\* (c) \*/

p3 = &nome[0];

aux = \*p3;

printf("(c) %c \n", aux);

/\* (d) \*/

p3 = &nome[4];

aux = \*p3;

printf("(d) %c \n", aux);

/\* (e) \*/

p3 = nome;

printf("(e) %c \n", \*p3);

/\* (f) \*/

p3 = p3 + 4;

printf("(f) %c \n", \*p3);

/\* (g) \*/

p3--;

printf("(g) %c \n", \*p3);

/\* <h> \*/

vetor[0] = 31;

vetor[1] = 45;

vetor[2] = 27;

p4 = vetor;

idade = \*p4;

printf("(h) %d \n", idade);

/\* (i) \*/

p5 = p4 + 1;

idade = \*p5;

printf("(i) %d \n", idade);

/\* (j) \*/

p4 = p5 + 1;

idade = \*p4;

printf("(j) %d \n", idade);

/\* (l) \*/

p4 = p4 - 2;

idade = \*p4;

printf("(l) %d \n", idade);

/\* (m) \*/

p5 = &vetor[2] - 1;

printf("(m) %d \n", \*p5);

/\* (n) \*/

p5++;

printf("(n) %d \n", \*p5);

return(0);

}

1. 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] = %.1f",i, vet[i]);

            printf("   \*(f + %d) = %.1f",i, \*(f+i));

            printf("   &vet[%d] = %X",i, &vet[i]);

            printf("   (f + %d) = %X",i, f+i);

      }

}

1. Assumindo que **pulo [10]** é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento da matriz?
   1. \*(pulo + 2)
   2. \*(pulo + 4)
   3. pulo + 4
   4. pulo + 2
2. Suponha a declaração: int mat[4], \*p, x; Quais expressões são válidas? Justifique:
   1. p = mat + 1;
   2. p = mat++;
   3. p = ++mat;
   4. x = ++ (\*mat);
3. 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("%X ",vet+i);    }  } |

1. O que fazem os seguintes programas quando executados?

|  |  |
| --- | --- |
| #include <stdio.h>  void main() {        int vet[] = {4,9,12};        int i,\*ptr;        ptr = vet;        for(i = 0 ; i < 3 ; i++) {             printf("%d ",\*ptr++);        }  } | #include <stdio.h>  void main(){        int vet[] = {4,9,12};        int i,\*ptr;        ptr = vet;        for(i = 0 ; i < 3 ; i++) {             printf("%d ",(\*ptr)++);        }  } |
| (a) | (b) |

1. Seja vet um vetor de 4 elementos: vet[4]. Supor que depois da declaração, vet esteja armazenado no endereço de memória 0060FF1C (ou seja, o endereço de vet[0]). Supor também que uma variável do tipo char ocupa 1 byte, do tipo int ocupa 4 bytes, do tipo float ocupa 4 bytes e do tipo double ocupa 8 bytes.

Qual o valor de vet+1, vet+2 e vet+3 se:

* 1. vet for declarado como char?
  2. vet for declarado como int?
  3. vet for declarado como float?
  4. vet for declarado como double?