**9.**  Escreva um programa que leia 3 números inteiros quaisquer e verifica se os valores lidos podem ser considerados uma tripla de Pitágoras, ou seja, a soma dos quadrados de dois números é igual ao quadrado terceiro. Caso tenhamos uma tripla de Pitágoras o seu programa imprime “eh uma tripla de Pitagoras” e caso não seja o seu programa imprime “nao eh tripla de Pitagoras”.

Exemplos:

**3 5 4**é uma tripla de Pitágoras

**5 3 4**é uma tripla de Pitágoras

**2 4 3** Não é tripla de Pitágoras

**10.**Escreva um programa que leia 3 valores de ponto flutuante e efetue o cálculo das raízes da equação de Bhaskara. Se não for possível calcular as raízes, mostre a mensagem correspondente “Impossivel calcular”, caso haja uma divisão por 0 ou raiz de número negativo.

**11.**(URI Online Judge | 1041) Escreva um programa que leia 2 valores com uma casa decimal (x e y), que devem representar as coordenadas de um ponto em um plano. A seguir, determine qual o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem (x = y = 0).

     Se o ponto estiver na origem, escreva a mensagem “Origem”.

Se o ponto estiver sobre um dos eixos escreva “Eixo X” ou “Eixo Y”, conforme for a situação.

**12.** (URI Online Judge | 1049) Neste problema, seu programa deverá ler 3 palavras que definem o tipo de animal possível segundo o esquema abaixo, da esquerda para a direita. Em seguida conclua qual dos animais seguintes foi escolhido, através das três palavras fornecidas

ofe

Exemplo: Se for informado ao seu programa **vertebrado**, **mamífero**, **onívoro** seu programa terá como saída **homem**

**1.**(URI ONLINE 1078) Leia um valor inteiro N (2 < N < 1000). A seguir, mostre a tabuada de N de 1 até 10, 1 x N = N      2 x N = 2N        ...       10 x N = 10N, por exemplo se for informado 140 o seu programa terá como saída:

1 x 140 = 140

2 x 140 = 280

3 x 140 = 420

4 x 140 = 560

5 x 140 = 700

6 x 140 = 840

7 x 140 = 980

8 x 140 = 1120

9 x 140 = 1260

10 x 140 = 1400

**2.** Faça um programa que 10 valores inteiros. Apresente então o maior valor lido e a posição dentre os 10 valores lidos.

**3.**Escreva um programa que calcule o MDC (máximo divisor comum) entre dois números naturais

**4.**Escreva um programa que lê um número natural, e verifica se o mesmo é primo ou não.

**5.**Escreva um programa para encontrar todos números primos existentes entre N1 e N2 (inclusive), em que N1 e N2 são números naturais lidos.

**6.**Escreva um programa que leia um número inteiro e positivo representando um número binário, determine o seu equivalente decimal. Exemplo: Dado 10010 a saída será 18, pois

1. 2 4 + 0. 2 3 + 0. 2 2 + 1. 2 1 + 0. 2 0 = 18

**7.**Escreva um programa que leia um número inteiro e positivo representando um número decimal, determine o seu equivalente binário. Exemplo: Dado 18 a saída deverá ser 10010.

**8.**Escreva um programa que receba um número inteiro N, esse número não contém o digito 0 (zero), e devolve N invertido, exemplo: se N igual 123 a resposta será 321.

**9.**Dizemos que um número natural *n* é *palíndromo*se*:*

1º algarismo de *n* é igual ao seu último algarismo,

2º algarismo de *n* é igual ao penúltimo algarismo e assim sucessivamente.

Exemplo: 567765 e 32423 são palíndromos

 567675 não é palíndromo.

Faça um programa que leia um número inteiro e positivo *n* e verifica se é palíndromo e imprime uma mensagem dizendo se o número lido é palíndromo ou não é palíndromo.

1) Escreva uma função que recebe dois vetores inteiros A[] e B[], em seguida, a sua função efetua a INTERSECÇÃO entre os vetores, ou seja, os elementos em comum entre os dois vetores, ao final sua função retorna uma String com a resposta. Os vetores dados não possuem valores duplicados e não estão ordenados.

            Exemplo:

A[] = { 7, 2, 5, 8, 4} e B[]= {4, 2, 9, 5} então A  B = {2, 5, 4}

            A[] = { 3, 9, 11} e B[]= {2, 6, 1} então A  B= {}

2) Repita o exercício anterior, agora deve ser retornado em uma String os elementos que estão em A[] mas não estão em B[], ou seja, a diferença de A – B, por exemplo:

            A[] = { 7, 2, 5, 8, 4} e B[]= {4, 2, 9, 5} então A – B = {7, 8 }

            A[] = { 3, 9, 11} e B[]= {2, 6, 1} então A –  B= {3, 9, 11}

3) Escreva uma função que recebe dois vetores inteiros A[] e B[], em seguida, a sua função efetua a UNIÃO entre os vetores, ou seja, os elementos do vetor A mais aos elementos do vetor B, desde que os elementos de B não estejam presentes no vetor A, ao final sua função retorna uma String com a resposta. Os vetores dados não possuem valores duplicados e não estão ordenados.

     Exemplo:

            A[] = { 7, 2, 5, 8, 4} e B[]= {4, 2, 9, 5} então A  B= {7, 2, 5, 8, 4, 9}

            A[] = { 3, 9, 11} e B[]= {2, 6, 1} então A  B = {3, 9, 11, 2, 6, 1}

4) A INTERCALAÇÃO é o processo utilizado para construir uma vetor ordenado crescentemente, de tamanho *n*+*m*, a partir de dois vetores já ordenados de tamanhos *n* e *m*. Por exemplo, a partir das sequências abaixo:

Exemplo:

A[] = { 1, 3, 6, 7} e B[] = {2, 4, 5}, o novo vetor C é feito a partir de A[] e B[]:

            C = { 1, 2, 3, 4, 5, 6, 7}

Escreva uma função que recebe dois vetores inteiros A[] e B[], em seguida, a sua função calcula e retorna o resultado da INTERCALAÇÃO, ou seja, o vetor C[].