**1- Faça um programa que carregue um vetor com 15 elementos inteiros e verifique a existência de elementos iguais a 30, mostrando as posições em que esses elementos apareceram.**

```
algoritmo
   decalre vetor, i numéricos

   parai de 0 a 14 faça
      leia vetor[i]
   fim-para

   parai de 0 a 14 faça
      teste(vetor[i], i)
   fim-para
fim-algoritmo

subrotina teste(numero, i numéricos)
   se(numero = 30)
      então imprima i
   fim-se
fim-subrotina
```

```c
#include <stdio.h>

void teste(int numero, int i);

int main() {
   int vetor[15];
   int i;

   for (i = 0; i < 15; i++) {
      printf("Insira o valor de vetor[%i]", i);
      scanf("%i", &vetor[i]);
   }

   for (i = 0; i < 15; i++)
      teste(vetor[i], i);

  printf("\n");
}

void teste(int numero, int i) {
   if(numero == 30)
      printf("\nElemento %i = 30", i);
}
```

**2- Leia um vetor de 50 posições de números inteiros e mostre somente os números positivos**

```
algoritmo
   decalre vetor[], i numéricos
   para i de 0 a 49 faça
      leia vetor[i]
   fim para

   para i de 0 a 49 faça
      teste(vetor[i])
   fim-para
fim-algoritmo

subrotina teste(numero numéricos)
   se(numero > 0)
     então imprima numero
   fim-se
fim-subrotina
```

```c
#include <stdio.h>

void teste(int numero);

int main() {
   int vetor[50];
   int i;

   for (i = 0; i < 50; i++) {
      printf("Insira o valor de vetor[%i]", i);
      scanf("%i", &vetor[i]);
   }

         printf("\nOs valores positivos do vetor sao: ");


   for (i = 0; i < 50; i++) {
      teste(vetor[i]);
   }

         printf("\n");
}

void teste(int numero) {
   if (numero > 0)
      printf("%i ", numero);
}
```

**3- Faça um programa que carregue uma matriz 10X3 com as notas de dez alunos em três provas. Mostre um relatório com o número do aluno (número da linha) e a prova em que cada aluno obteve a menor nota. Ao final do relatório, mostre quantos alunos tiveram menor nota na prova 1, quantos alunos tiveram menor nota na prova 2 e quantos alunos tiveram menor nota na prova 3.**

```
algoritmo
   decalre matriz[][], i, j, m1, m2, m3 numéricos

   para i de 0 a 9 faça
      para j de 0 a 2 faça
         leia matriz[i][j]
      fim-para
   fim-para

   para i de 0 a 9 faça
     teste(matriz[i][0], matriz[i][1], matriz[i][2], i, &m1, &m2, &m3)
   fim-para

   imprima m1, m2, m3
fim-algoritmo

subrotina teste(nota1, nota2, nota3, i, *m1, *m2, *m3)
   imprima i

   se(nota1 < nota2 && nota 1 < nota3)
      *m1++
      imprima "Prova 1"
   fim-se

   se(nota2 < nota1 && nota2 < nota3)
      *m2++
      imprima "Prova 2"
   fim-se

   se(nota3 < nota1 && nota3 < nota2)
      *m3++
      imprima "Prova 3"
   fim-se
fim-subrotina
```

```c
#include <stdio.h>

void teste(int nota1, int nota2, int nota3, int i, int* m1, int* m2, int* m3);

int main() {
   int matriz[10][3];
   int i, j, m1, m2, m3;

          m1 = 0;
          m2 = 0;
          m3 = 0;

   for (i = 0; i < 10; i++) {
      for (j = 0; j < 3; j++) {
         printf("Insira a nota %i do aluno %i ", j, i);
         scanf("%i", &matriz[i][j]);
      }
   }

   for (i = 0; i < 10; i++)
      teste(matriz[i][0], matriz[i][1], matriz[i][2], i, &m1, &m2, &m3);

   printf("\nMenor nota na Prova 1: %i alunos\nMenor nota na Prova 2: %i alunos\nMenor nota na Prova 3: %i
alunos\n", m1, m2, m3);
}

void teste(int nota1, int nota2, int nota3, int i, int* m1, int* m2, int* m3) {
   printf("\nAluno %i: ", i);

   if(nota1 < nota2 && nota1 < nota3) {
      m1 = m1+1;
      printf("Menor nota na Prova 1");
   }

   if(nota2 < nota1 && nota2 <= nota3) {
      m2 = m2+1;
      printf("Menor nota na Prova 2");
   }

   if(nota3 < nota2 && nota3 < nota1) {
      m3 = m3+1;
      printf("Menor nota na Prova 3");
   }
}
```

**4- Escreva um programa que leia um vetor de 10 posições e mostre-o. Use uma subrotina que inverta o vetor, trocando o 1º elemento com o último, o 2º elemento com o penúltimo e assim sucessivamente. Mostre o novo vetor depois da troca.**

```
algoritmo
   decalre vetor[], i numéricos
   para i de 0 a 9
      leia vetor[i]
   fim-para

   para i de 0 a 4
      troca(&vetor[], i)
   fim-para

   para i de 0 a 9
       imprima vetor[i]
   fim-para
fim-algoritmo

subrotina troca(*vetor[], i numéricos)
   decalre aux
   aux = vetor[i]
   vetor[i] = vetor[9-i]
   vetor[9-i] = aux
fim-subrotia
```

```c
#include <stdio.h>

void troca(int* vetor, int i);

void main() {
   int vetor[10], i;

   for(i = 0; i < 10; i++) {
      printf("Insira o valor de vetor[%i]", i);
      scanf("%i", &vetor[i]);
   }

   for(i =0; i < 4; i++)
      troca(vetor, i);

   for(i = 0; i < 10; i++)
      printf("%i ", vetor[i]);

         printf("\n");
}

void troca(int* vetor, int i) {
    int aux = vetor[i];
    vetor[i] = vetor[9-i];
    vetor[9-i] = aux;
}
```

**5 - Faça um programa que use uma subrotina parareceber como parâmetro dois vetores, contendo valores da coordenada x e valores da coordenada y de pontos no plano cartesiano. A subrotina deve calcular os coeficientes a e b de uma reta y=ax+b que é a regressão linear dos pontos.**

```
algoritmo
   decalre vetorX[], vetorY[], i, tamanho numericos
   leia tamanho

   parai de 0 a tamanho-1
      leia vetorX[i]
      leia vetorY[i]
   fim-para

   calcular(vetorX, vetorY, tamanho)
fim-algoritmo

subrotina calcular(vetorX[], vetorY[], tamanho numéricos)
   decalre somaXY, somaX, somaY, somaX2 numericos

   parai de 0 a tamanho-1
      somaXY = somaXY+(vetorX[i] * vetorY[i])
      somaX = somaX + vetorX[i]
      somaX2 = somaX2 + pow(vetorX[i], 2)
      somaY = somaY + vetorY[i]
   fim-para

   a = ((tamanho*somaXY)-(somaX*somaY))/((b*somaX2)-pow(somaX,2))
   b = ((somaY*somaX2)-(somaX*somaXY))/((n*somaX2)-pow(somaX, 2))

   imprima a, b
fim-subrotina
```

```
#include <stdio.h>
#include <math.h>
#define TAMANHO 10

void calcular(float* vetorX, float* vetorY);

void main() {
   float vetorX[TAMANHO], vetorY[TAMANHO];
   int i;

    for(i = 0; i < TAMANHO; i++) {
      printf("Insira o valor de X, Y parap%i(x, y)", i);
      scanf("%f, %f", &vetorX[i], &vetorY[i]);
   }

   calcular(vetorX, vetorY);
}

void calcular(float* vetorX, float* vetorY) {
   float somaXY, somaX, somaY, somaX2, a, b, tamanho;
         int i = 0;

   for (i = 0; i < TAMANHO; i++) {
      somaXY = somaXY+(vetorX[i] * vetorY[i]);
      somaX = somaX + vetorX[i];
      somaX2 = somaX2 + pow(vetorX[i], 2);
      somaY = somaY + vetorY[i];
   }


   a = ((TAMANHO*somaXY)-(somaX*somaY));
   a = a/((TAMANHO*somaX2)-pow(somaX,2));
   b = ((somaY*somaX2)-(somaX*somaXY));
   b = b/((TAMANHO*somaX2)-pow(somaX, 2));

   printf("Na regressão linear do plano cartesiano apresentado dada pela fórmula y=ax+b\n
            Os coeficientes a = %f e b = %f", a , b);
}
```