

## ImprimeImpares

Escreva uma função para imprimir os  $n$  primeiros números naturais ímpares.

Faça a leitura de um número natural  $n$  e a chamada da função criada na função main.

Exemplo:

Entrada:

4

Saída:

1 3 5 7

## NumeroTriangular

Escreva uma função que receba um inteiro positivo  $n$ , e verifique se  $n$  é triangular, ou seja, se  $n$  é igual ao produto de 3 números inteiros consecutivos. Por exemplo, 120 é triangular pois  $4 \times 5 \times 6 = 120$ . A função deve retornar 1 se  $n$  é triangular, ou 0, caso contrário. Imprima todos os números triangulares entre 1 e 1000, fazendo a chamada da função criada a partir da função principal.

## InverteVetor

Escreva uma função que receba um vetor de inteiros e o tamanho do vetor e inverta os elementos do vetor.

Na função principal, crie um vetor de tamanho  $n$ , faça a leitura dos valores e chame a função criada.

Imprima o vetor após a chamada da função.

Exemplo:

Entrada:

1

2

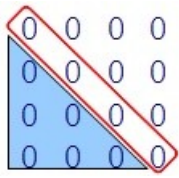
3

Saída:

3 2 1

## SomaAbaixoDiagonal

Escreva uma função que receba uma matriz de inteiros de tamanho  $n \times n$  e o valor de  $n$  e retorne a soma dos elementos abaixo da diagonal principal (excluindo-se a diagonal principal - área em azul na figura abaixo).



Exemplo de teste:

Entrada:

0 1 0

2 4 7

1 3 5

Saída:

6

## MaiorComprimento

Escreva uma função que receba uma string (vetor de caracteres) contendo uma frase e retorne o tamanho (número de caracteres) da palavra de maior comprimento da frase. Considere que a string de entrada possui no máximo 100 caracteres e não contém pontuações.

Exemplo de teste:

Entrada:

Boa tarde

Saída:

5

## MatrizLinCol

Escreva uma função que receba uma matriz  $n \times n$  de inteiros e um inteiro não negativo  $n$ , e verifique se a matriz satisfaz a seguinte regra:

- cada linha e cada coluna contém  $n-1$  elementos nulos e um único elemento igual a 1.

A função retorna 1, em caso positivo, ou 0, caso contrário.

Exemplo de teste:

Entrada:

0 1 0 0

0 0 1 0

1 0 0 0

0 0 0 1

Saída:

1

## MinMax

Escreva uma função que recebe um vetor de inteiros e o tamanho do vetor e retorne os valores mínimo e máximo do vetor. Esses valores (mínimo e máximo) devem ser devolvidos através de parâmetros passados por referência.

Exemplo de teste:

Entrada:

3

5 0 -2

Saída

-2 5

## NumeroDeOcorrencias

Escreva uma função que receba uma cadeia de caracteres s e um caractere x e retorne o número de ocorrências de x em s. Considere que a cadeia de caracteres possui no máximo 100 caracteres.

Obs.: utilize um vetor de caracteres alocado dinamicamente. Utilize a notação de ponteiro (em vez de colchetes) para manipular o vetor.

Exemplo de teste:

Entrada:

disciplina

i

Saída

3

## Primos

Escreva uma função que recebe dois inteiros a e b e retorne um vetor (um ponteiro para o início do vetor) contendo todos os números primos entre a e b (inclusive).

Exemplo de teste:

Entrada:

1 5

Saída

2 3 5

## PontoRetangulo

Escreva uma função para verificar se um ponto p(x,y) está dentro de um retângulo, dados os vértices inferior esquerdo r1(x,y) e superior direito r2(x,y). A função deve receber 3 pontos e retornar 1, caso o ponto esteja dentro do retângulo, ou 0, caso contrário.

Utilize a estrutura a seguir para a representação do ponto:

```
struct ponto {  
    float x;
```

```
float y;  
}
```

Exemplo de teste:

Entrada (p, r1, r2):

2 2

1 1

3 4

Saída:

1

## PontosRetangulo

Modifique o programa anterior de modo que dada uma sequência de n pontos, apresente quantos pontos estão dentro de um retângulo. Acrescente uma função para fazer essa verificação. Esta função recebe um vetor de pontos, o seu tamanho e os dois vértices do retângulo e retorna um inteiro.

Obs.: utilize um vetor alocado dinamicamente. Utilize a notação de ponteiro (em vez de colchetes) para manipular o vetor.

Exemplo de teste:

Entrada (n, seq. de pontos, r1, r2):

3

1 2

3 3

-1 2

0 0

5 6

Saída:

2

## ConcatenaStrings

Escreva uma função que recebe duas cadeias de caracteres s1 e s2, e retorne uma cadeia de caracteres s3 que é a concatenação de s1 com s2.

Utilize a seguinte declaração para a função:

```
char *concatena(char *s1, char *s2)
```

Exemplo de teste:

Entrada:

algo

ritmo

Saída

algoritmo

## ListaAlunos

Considere a estrutura abaixo para representar os dados de um aluno em uma disciplina (RA, nome, notas das provas e atividades). Escreva uma função que receba um vetor de alunos e o seu tamanho e imprima os dados de cada aluno (RA e nome), a sua média final e a situação (aprovado ou reprovado). Para o cálculo da média considere os pesos: 35% para cada prova e 30% para atividades. Para aprovação, é necessário ter média pelo menos 5.0. Apresente também o número de aprovados e reprovados no final da lista (veja o exemplo de teste abaixo).

Obs.: Apresente a média com apenas duas casas decimais depois da vírgula.

```
struct aluno {  
    int ra;  
    char nome[20];  
    float p1;  
    float p2;  
    float atv;  
}
```

Exemplo de teste:

Entrada (núm. de alunos, RA, nome e 3 notas para cada aluno):

```
2  
123  
Abc  
6 7 8  
135  
Bcd  
3 4 5  
221
```

Saída:

```
123 Abc 6.95 Aprovado  
221 Bcd 3.95 Reprovado  
Aprovados: 1  
Reprovados: 1
```

## MediaVarianciaDesvioPadrao

Modifique o programa do exercício anterior para incluir uma função que calcule a média geral da turma, a variância e o desvio padrão. Esta função recebe um vetor de alunos e o seu tamanho e retorna uma estrutura, conforme abaixo, contendo os valores calculados.

Obs.: Na saída, apresente os valores com apenas duas casas decimais depois da vírgula.

```
struct medidas {  
    float media;
```

```
float variancia;  
float desvio_padrao;  
}
```

Entrada (núm. de alunos, RA, nome e 3 notas para cada aluno):

2

123

Abc

6 7 8

135

Bcd

3 4 5

221

Saída:

123 Abc 6.95 Aprovado

221 Bcd 3.95 Reprovado

Aprovados: 1

Reprovados: 1

Media: 5.45

Variancia: 2.25

Desvio padrao: 1.50