

Ex. 1 Escreva uma função de nome `soma_prod()` com quatro argumentos: v, n, s, p . Os argumentos v e n são entradas para a função, e s e p são argumentos de saída. v é um vetor de inteiros, n é o tamanho deste vetor, s é a soma dos elementos do vetor e p é seu produto. Escreva um programa em C++ que lê um vetor de N elementos inteiros a partir do teclado, e imprime a soma e o produto dos elementos do vetor, calculados com a função `soma_prod()`. N é definido com um `#define`.

A função deve computar dois resultados e o mecanismo a ser usado para retornar dois valores consiste em passar o endereço das duas variáveis que receberão os valores dos dois resultados, que neste caso são s e p .

Ex. 2 Dado um vetor com 1000 elementos inteiros (`vals[]`), cada elemento contendo um número entre 0 e 99 (inclusive), escreva uma função que recebe o vetor `vals[]` e um vetor chamado `histo[]`, e preenche este último com o “histograma” de `vals[]`. Por “histograma” entende-se a contagem de vezes em que o valor inteiro i ($0 \leq i \leq 99$) ocorre em `vals[]`. Por exemplo, se o número N aparece 300 vezes em `vals[]`, então `histo[N]` deve ser preenchido com 300. Por outro lado, se N não aparece em `vals[]` então `histo[N]` deve ser preenchido com zero.

Ex. 3 Escreva uma função na Linguagem C++, chamada `maxmin()`, que possui 2 argumentos: `mat` e `mm`. O argumento `mat` é uma matriz de M linhas e N colunas e o argumento `mm` é uma matriz de 2 linhas e 2 colunas. Os valores M e N devem ser definidos através da diretiva `#define`. A função deve encontrar o maior e menor valores existentes em `mat`, e quantas vezes cada um deles ocorre em `mat`. O menor valor e o número de vezes que ele ocorre deve ser colocado na primeira linha da matriz `mm`, respectivamente na primeira e segunda coluna. O maior valor e o número de vezes que ele ocorre deve ser colocado na segunda linha da matriz `mm`, respectivamente na primeira e segunda coluna. Suponha que a função `maxmin()` receba em `mat` a matriz abaixo ($M = 3$ e $N = 4$); seu programa deve produzir a matriz da direita.

1 2 3 5	Ao final da execução da função, a matriz <code>mm</code> deverá ter o con-	0 1
2 7 2 6	teúdo abaixo, indicando que o menor valor na matriz é 0 e	7 3
0 3 7 7	ocorre 1 vez, e o maior valor é 7 e ocorre 3 vezes:	

A solução para este problema deve empregar a estratégia de “dividir para conquistar”: resolve-se um problema simples de cada vez. Inicialmente a matriz é percorrida na busca dos valores máximo e mínimo, para então contar as ocorrências dos máximo e mínimo. Note que estes dois passos devem ser executados em seqüência – não é possível contar as ocorrências do máximo (mínimo) até que este seja encontrado.

A busca pelos máximo (mínimo) inicia com a suposição de que `mat[0][0]` é o valor máximo (mínimo) – poderia ser qualquer outro elemento – e então percorre-se a matriz efetuando uma troca a cada “novo” máximo (mínimo) que é encontrado.