

**Problema 01) Números inteiros**

Escreva um programa em C que ordena números inteiros.

Entrada: A primeira linha contém **n** ( $n > 0$ ) a quantidade de números a ordenar. A segunda linha possui os números a ordenar, separados por um espaço em branco entre si.

Saída: Os números ordenados, separados por um espaço em branco entre si.

Entrada	Saída
7 5 10 20 3 2 7 11	2 3 5 7 10 11 20

**Problema 02) Números racionais**

Escreva um programa em C que ordena números racionais, contudo considerando somente a parte inteira do número para o algoritmo de ordenação. A ordenação deve ser estável.

Entrada: A primeira linha contém **n** ( $n > 0$ ) a quantidade de números a ordenar. A segunda linha possui os números a ordenar, separados por um espaço em branco entre si.

Saída: Os números ordenados, apresentando um número em cada linha, apresentando duas casas decimais.

Entrada	Saída
7 5.21 10.02 5.1 3.7 2.7 3.5 5.05	2.70 3.70 3.50 5.21 5.10 5.05 10.02

**Problema 03) Selection sort**

Faça um algoritmo implementando o *Selection Sort* (ordenação por seleção) e, para cada troca no vetor, imprima os elementos que foram trocados e, por fim, imprima o vetor ordenado. Se não houver troca de elementos, imprima somente o vetor ordenado. Observe que a implementação deve ser estável, com mínimo de trocas.

Entrada: A primeira linha da entrada contém um inteiro **n** ( $n > 0$ ) que corresponde ao tamanho do vetor. A segunda linha da entrada contém os **n** elementos (números inteiros) do vetor.

Saída: Seu programa deve imprimir as trocas ocorridas e o vetor ordenado.

Entrada	Saída
5 5 4 3 2 1	5 <-> 1 4 <-> 2 1 2 3 4 5
6 4 2 4 1 7 2	4 <-> 1 4 <-> 2 7 <-> 4 1 2 2 4 4 7

**Problema 04) Insertion sort**

Repita o problema 03, mas utilizando o *Insertion Sort* (ordenação por inserção). Observe que a implementação deve ser estável, com mínimo de trocas.

Entrada: A primeira linha da entrada contém um inteiro  $n$  ( $n > 0$ ) que corresponde ao tamanho do vetor. A segunda linha da entrada contém os  $n$  elementos (números inteiros) do vetor.

Saída: Seu programa deve imprimir as trocas ocorridas e o vetor ordenado.

Entrada	Saída
5 5 4 3 2 1	5 <-> 4 5 <-> 3 4 <-> 3 5 <-> 2 4 <-> 2 3 <-> 2 5 <-> 1 4 <-> 1 3 <-> 1 2 <-> 1 1 2 3 4 5
6 4 2 4 1 7 2	4 <-> 2 4 <-> 1 4 <-> 1 2 <-> 1 7 <-> 2 4 <-> 2 4 <-> 2 1 2 2 4 4 7

**Problema 05) Bubble sort**

Repita o problema 03, mas utilizando o *Bubble Sort* (ordenação por flutuação, ordenação por bolha). Observe que a implementação deve ser estável, com mínimo de trocas.

Entrada: A primeira linha da entrada contém um inteiro  $n$  ( $n > 0$ ) que corresponde ao tamanho do vetor. A segunda linha da entrada contém os  $n$  elementos (números inteiros) do vetor.

Saída: Seu programa deve imprimir as trocas ocorridas e o vetor ordenado.

Entrada	Saída
5 5 4 3 2 1	5 <-> 4 5 <-> 3 5 <-> 2 5 <-> 1 4 <-> 3 4 <-> 2 4 <-> 1 3 <-> 2 3 <-> 1 2 <-> 1 1 2 3 4 5
6 4 2 4 1 7 2	4 <-> 2 4 <-> 1 7 <-> 2 4 <-> 1 4 <-> 2 2 <-> 1

	4 <-> 2 1 2 2 4 4 7
--	------------------------

**Problema 06) Ordena-soma**

Dados dois inteiros **p** e **q**, escreva um programa em C para imprimir a soma dos elementos de um vetor de números inteiros, considerando apenas o intervalo definido pelo p-ésimo menor elemento até o q-ésimo menor elemento do vetor.

Entrada: A primeira linha da entrada contém um inteiro **n** ( $n > 0$ ), que corresponde ao tamanho do vetor. A segunda linha da entrada possui os **n** elementos do vetor, separados por um espaço em branco entre si. Por fim, a terceira linha da entrada contém os valores de **p** e **q** ( $0 < p \leq q \leq n$ ).

Saída: Seu programa deve imprimir uma única linha, com a soma dos elementos no intervalo.

Entrada	Saída
5 5 4 3 2 1 1 3	6
6 4 2 4 1 7 2 2 5	12

**Problema 07) Máquina de Café**

No dia da Independência, Pedro decidiu distribuir cafés grátis. Ele instalará máquinas de café em uma cafeteria, cada máquina demora um minuto para servir uma pessoa. Ele recebeu a lista de **n** pessoas que visitarão sua cafeteria. O tempo de visita de cada pessoa é denotado por dois inteiros (**h**, **m**) onde **h** é a hora da visita e **m** é o minuto da visita naquela hora.

Pedro quer configurar o número mínimo de máquinas de café, para que ninguém espere ‘desocupar’ alguma máquina para ser atendido. Ajude-o a encontrar o número mínimo de máquinas de café necessário para que não haja espera.

Entrada: A primeira linha contém **t** ( $1 \leq t \leq 100$ ), o número de casos de testes. Cada caso de teste refere-se a um grupo de linhas. A primeira linha de cada caso de teste contém **n** ( $1 \leq n \leq 1000$ ), o número de pessoas a serem servidas naquele dia. As próximas **n** linhas contém dois inteiros (**h** e **m**,  $0 \leq h \leq 23$ ,  $0 \leq m \leq 59$ ) separados por um espaço em branco, referentes a hora e minuto de cada cliente, respectivamente

Saída: Para cada caso de teste, um inteiro denotando o número mínimo de máquinas de café.

**Exemplo**

Entrada	Saída
2 5 10 20 10 20 10 30 10 20 10 30 4 10 20 5 40 10 21 23 11	3 1

1 8 10 22 10 24 10 21 10 22 10 20 10 22 10 21 10 20	3
--	---

**Problema 08) Máquina de Café 2**

O problema anterior foi modificado, pois cada máquina de café instalada demora dois minutos para servir uma pessoa. Ajude a encontrar o número mínimo de máquinas de café necessário para que não haja espera.

Entrada: A primeira linha contém  $t$  ( $1 \leq t \leq 100$ ), o número de casos de testes. Cada caso de teste refere-se a um grupo de linhas. A primeira linha de cada caso de teste contém  $n$  ( $1 \leq n \leq 1000$ ), o número de pessoas a serem servidas naquele dia. As próximas  $n$  linhas contém dois inteiros ( $h$  e  $m$ ,  $0 \leq h \leq 23$ ,  $0 \leq m \leq 59$ ) separados por um espaço em branco, referentes a hora e minuto de cada cliente, respectivamente

Saída: Para cada caso de teste, um inteiro denotando o número mínimo de máquinas de café.

**Exemplo**

Entrada	Saída
2 5 10 20 10 20 10 30 10 20 10 30 4 10 20 5 40 10 21 23 11	3 2
1 8 10 22 10 24 10 21 10 22 10 20 10 22 10 21 10 20	5

**Problema 09) Olimpíadas**

O Comitê Olímpico Internacional (COI) está visitando as cidades candidatas a sediar as Olimpíadas em 2020. O COI decidiu premiar com medalha os quatro primeiros colocados em cada modalidade (ouro, prata, bronze e latão).

Sua tarefa é escrever um programa que, dada a informação dos países que receberam medalhas, indicar a classificação de cada dos países; cada país é identificado por um número inteiro.

O melhor colocado deve ser o país que conseguiu o maior número de medalhas de ouro. Se houver empate entre países no número de medalhas de ouro, o desempate é resolvido pelo maior número de medalhas de prata. Se houver empate também no número de medalhas de prata, o desempate é resolvido pelo maior número de medalhas de bronze. Se houver empate também no número de medalhas de bronze, o desempate é resolvido pelo maior número de medalhas de latão. Se ainda assim houver empate entre dois ou mais países, o melhor classificado é o que possui o maior número de identificação (ufa... ).

**Entrada:** A primeira linha da entrada contém dois números inteiros **n** e **m**, separados por um espaço em branco, indicando respectivamente o número de países ( $4 \leq n \leq 100$ ) e número de modalidades esportivas envolvidas na competição ( $1 \leq m \leq 100$ ). Os países são identificados por números inteiros de 1 a **n**.

Cada uma das **m** linhas seguintes contém quatro números inteiros **o**, **p**, **b** e **l**, separados por um espaço em branco, representando a identificação dos países cujos atletas receberam respectivamente medalhas de ouro, prata, bronze e latão. Por exemplo, a entrada “3 4 2 5” indica que os países que receberam as medalhas de ouro, prata, bronze e latão são identificados por 3, 4, 2 e 5, respectivamente.

**Saída:** Uma única linha contendo **n** números, separados por um espaço em branco, representando os países na ordem decrescente de classificação (o primeiro número representa o país que é o primeiro colocado, o segundo número representa o país que é o segundo colocado, e assim por diante).

Exemplo

Entrada	Saída
7 3 2 6 2 1 1 3 2 3 7 1 2 4	1 2 7 3 6 4 5
7 3 5 1 2 4 6 3 4 1 7 2 3 4	7 6 5 3 2 1 4

**Problema 10) Olimpíadas 2**

O problema 09 foi modificado. As medalhas de ouro, prata, bronze e latão valem, respectivamente, 4, 3, 2 e 1 pontos. O melhor colocado é o país que tiver o maior número de pontos. Em caso de empate, o melhor classificado é o que possui o maior número de identificação.

Exemplo

Entrada	Saída
7 3 2 6 2 1 1 3 2 3 7 1 2 4	2 1 7 3 6 4 5
7 3 5 1 2 4 6 3 4 1 7 2 3 4	3 2 7 6 5 4 1