INSTITUTO FEDERAL
Paraíba
Campus Guarabira

LACED
Laboratório de Computação Embarcada e Distribuída

ORI
Olimpíada Regional de Informática

MODALIDADE PROGRAMAÇÃO
2ª FASE

A PROVA TERÁ DURAÇÃO DE **DUAS HORAS E TRINTA MINUTOS**

Este Caderno contém 5 problemas
04 de Outubro de 2019

Instruções

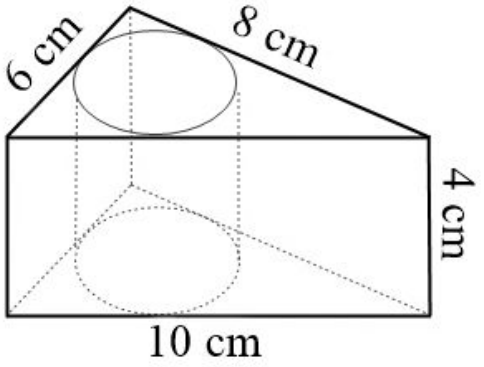
LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto de 8 páginas . Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a internet, livros, anotações ou qualquer outro material durante a prova.
- É permitida a consulta ao menu ajuda do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas não estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver as mais fáceis primeiro.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo .c ; soluções na linguagem C++ devem ser arquivos com sufixo .cpp; soluções na linguagem Java devem ser arquivos com sufixo . java e a classe principal deve ter o mesmo nome do arquivo fonte; e soluções na linguagem Python 2.7 devem ser arquivos com sufixo .py2 e para a linguagem Python 3.6 o sufixo é .py3.
- Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entradas e saída de dados:
    - em C: scanf, getchar, printf, putchar;
    - em Pascal: readln, read, writeln, write;
    - em C++: as mesmas do C ou os objetos cout e cin;
    - em Java: qualquer classe ou função padrão, como por exemplo Scanner, BufferedReader, BufferedWriter e System.out.println;
    - em Python 2.7 ou Python 3.6 : read, readline, readlines, print, write, raw_input e input;
    - **a correção é feita de forma automática. Então, certifique-se de remover qualquer caractere de *return* e *newline* dos dados de entrada caso ache necessário.**
- As versões dos compiladores e parâmetros de comandos utilizados na correção são:
    - em C: Clang 8.0.0 com os parâmetros -static -O2 -lm;
    - em C++: GCC 8.3.0 com os parâmetros -static -O2 -std=c++17 -lm;
    - em Java: OpenJDK 12;
    - em Pascal: Free Pascal 3.0.4;
    - em Python 2: Python 2.7.16;
    - em Python 3: Python 3.6.9.
- Procure resolver o problema de maneira eficiente. Na correção, Eficiência também será levada em conta. Em cada problema é indicado o tempo limite. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.
- Para a realização da prova entre no endereço <https://boca.laced.com.br/>, faça seu login e boa sorte.

Problema A. Bolo da Juju

Nome do Programa: A.(c | pas | cpp | java | py)
Tempo: 1 segundo

Juju é uma senhora muito criativa. Em sua loja de bolos, ela inventa bolos de tamanhos e sabores variados para os amantes da geometria. Atualmente, ela criou um novo modelo de bolo: um prisma reto com base triangular, vazado de tal maneira que a perfuração na forma seja no formato um cilindro circular reto sendo tangente às suas faces laterais, que serve para armazenar o recheio do bolo, como no exemplo da figura:



Ajude Juju a saber o qual a capacidade que o cilindro circular reto comporta de recheio usando um programa. **Considere pi = 3.14**.

Entrada

A entrada é composta por uma única linha, contendo 4 valores inteiros, d1, d2, d3 e h (1 <= d1, d2, d3, h <=10.000), representando respectivamente as 3 dimensões da base e por último a altura.

Saída

A saída é a capacidade do cilindro, um número com 2 dígitos após o ponto decimal.

Casos de Teste

Entrada	Saída
6 8 10 10	125.60

Entrada	Saída
12 16 20 8	401.92

Problema B. Frequência de funcionários

Nome do Programa: B.(c | pas | cpp | java | py)
Tempo: 1 segundo

No laboratório responsável por elaborar as provas da Olimpíada Regional de Informática (ORI) há muitos funcionários que criam as questões e as armazenam em um banco de dados. O local é protegido por um sistema de segurança usando reconhecimento facial que ao identificar e reconhecer o funcionário libera a sua entrada e registra o seu nome em um outro banco de dados de entradas durante o dia.
Sua tarefa neste problema é receber os dados de entrada dos funcionários e, seguidamente, dizer ao usuário quantas vezes cada funcionário aparece na entrada de dados, escrevendo a frequência em ordem alfabética.

Entrada

A entrada contém apenas 1 caso de teste. A primeira linha de entrada contém um único inteiro N (1<N<3000), que indica a quantidade dados que serão lidos logo em seguida.

Saída

Imprima para o usuário a saída indicando o nome em ordem alfabética, seguida da palavra "apareceu", depois quantas vezes aparece e a palavra "vez(es)", seguindo os exemplos de casos de testes.

Casos de teste

Entrada	Saída
5 Joyce Tania Joyce Jose Jose	Jose apareceu 2 vez(es) Joyce apareceu 2 vez(es) Tania apareceu 1 vez(es)

Entrada	Saída
6 Jose Emanuel Gabriele Antonio Ana Gabriele	Ana apareceu 1 vez(es) Antonio apareceu 1 vez(es) Emanuel apareceu 1 vez(es) Gabriele apareceu 2 vez(es) Jose apareceu 1 vez(es)

Problema C. Festas do ano

Nome do Programa: C.(c | pas | cpp | java | py)
Tempo: 1 segundo

A família de Ava sempre realiza algumas comemorações durante o ano, mais especificamente, o Ano Novo, São João e Halloween, além do aniversário de Ava e de seu irmão Ivo. Para saber quantos dias faltam até essas comemorações do atual e do próximo ano, a família dela contratou-lhe para resolver esse problema usando programação. **Atente-se aos anos bissextos.**

Entrada

A entrada é composta por três linha contendo na primeira o dia, mês e ano do dia atual, na segunda o dia e o mês do aniversário de Ava e na terceira o dia e o mês do aniversário de Ivo.

Saída

A saída é composta por 10 linhas (contendo a quantidade de dias que faltam para a data no ano atual e no próximo, respectivamente na sequência: Ava, Ivo, Ano Novo, São João e Halloween) contendo a mensagem “passou” caso o dia já estiver passado ou a quantidade de dias separado por espaço em branco seguido da palavra “days”, uma vírgula e outro espaço em branco e a hora, minutos e segundos do início do dia (meia noite), separado por dois pontos. Se o dia atual corresponder a alguma das datas, imprima apenas “0:00:00” ao invés de todos os dados da data.

Caso de teste

Entrada	Saída
25 8 2019 16 5 9 1	passou 265 days, 0:00:00 passou 137 days, 0:00:00 passou 129 days, 0:00:00 passou 304 days, 0:00:00 67 days, 0:00:00 433 days, 0:00:00

Problema D. Enigma 6174

Nome do Programa: D.(c | pas | cpp | java | py)
Tempo: 1 segundo

O enigma do número 6174, conhecido também como Constante de Kaprekar, em homenagem ao seu descobridor, o indiano D. R. Kaprekar, tem assustado os matemáticos há décadas. O processo inicia ao escolher um número de quatro dígitos com ao menos dois de seus dígitos diferentes, incluindo o zero, depois cria-se dois novos números, um com a ordenação dos dígitos em ordem crescente e outro em ordem decrescente, em seguida, subtrai-se o maior dos números pelo menor, repetindo os três últimos passos até que o resultado da subtração seja igual a 6174.

Entrada

A entrada é composta por uma única linha de código contendo um número inteiro com 4 dígitos sendo dois desses diferentes entre si.

Saída

A saída é composta por várias linhas contendo o resultado das operações de subtração entre os números e por fim a quantidade de vezes em que a operação ocorreu até que o número resultasse na constante.

Casos de teste

Entrada	Saída
1234	3087 8352 6174 3

Entrada	Saída
1945	8082 8532 6174 3

Entrada	Saída
2356	4176 6174 2

Problema E. Onde está o Mármore?

Nome do Programa: E.(c | pas | cpp | java | py)
Tempo: 2 segundos

Raju e Meena adoram jogar um jogo diferente com pequenas peças de mármores, chamados Marbles. Eles têm um monte destas peças com números escritos neles. No início, Raju colocaria estes pequenos mármores um após outro em ordem ascendente de números escritos neles. Então Meena gostaria de pedir a Raju para encontrar o primeiro mármore com um certo número. Ele deveria contar 1...2...3. Raju ganha um ponto por cada resposta correta e Meena ganha um ponto se Raju falha. Depois de um número fixo de tentativas, o jogo termina e o jogador com o máximo de pontos vence. Hoje é sua chance de jogar com Raju. Sendo um(a) pessoa esperto(a), você tem em seu favor o computador. Mas não subestime Meena, ela escreveu um programa para monitorar quanto tempo você levará para dar todas as respostas. Portanto, agora escreva o programa, que ajudará você em seu desafio com Raju.

Entrada

A entrada contém vários casos de teste, mas o total de casos é menor do que 65. Cada caso de teste inicia com dois inteiros: **N** que é o número de mármores e **Q** que é o número de consultas que Meena deseja fazer. As próximas **N** linhas conterão os números escritos em cada um dos **N** mármores. Os números destes mármores não tem qualquer ordem em particular. As seguintes **Q** linhas irão conter **Q** consultas. Tenha certeza, nenhum dos números da entrada é maior do que 10000 e nenhum deles é negativo.

A entrada é terminada por um caso de teste onde **N** = 0 e **Q** = 0.

Saída

Para cada caso de teste de saída deve haver um número serial do caso de teste. Para cada consulta, escreva uma linha de saída. O formato desta linha dependerá se o número consultado estiver ou não escrito em um dos mármores. Os dois diferentes formatos são descritos abaixo:
'x found at y', se o primeiro marble x foi encontrado na posição y. Posições são numeradas de 1, 2,... a N.
'x not found', se o marble com o número x não estiver presente.

Casos de Teste

Entrada	Saída
4 1 2 3 5 1 5 5 2 1 3 3 3 1 2 3 0 0	CASE# 1: 5 found at 4 CASE# 2: 2 not found 3 found at 3