

MARATONA DE PROGRAMAÇÃO

SEMANA DO IME

25/10/2023 13:00 às 17:00

Sala 6142-F

Caros participantes: esse documento com as informações sobre a Maratona de Programação da Semana do IME/2023 dadas a seguir deve ser lido com bastante atenção.

1. Os alunos trabalharão isoladamente ou em duplas, usando um computador apenas. Isso será decidido junto com a Organização, no início da maratona.
Chegue 15min antes do horário de início para seu cadastramento!

2. Usaremos o site Beecrowd para hospedar a competição. Para se habilitar a participar, você deve fazer o seguinte:

a) **Registrar-se no site Beecrowd:** você deve criar um usuário nesse site.
Acesse o link

<https://www.beecrowd.com.br>

e faça seu cadastro. Atenção: traga suas credenciais anotadas para o dia da competição, para que não tenha problemas de acesso!

b) **Registrar-se na Competição:** A competição consiste de uma atividade (Homework) dentro de uma "disciplina" do Beecrowd. Para se registrar, siga os passos abaixo:

- logado no site Beecrowd, em sua tela inicial, clicar em "Academic" (o botão que aparece na parte superior da página);
- Clicar em "Acessar Disciplina" (botão azul que na parte superior e à direita da página);
- Fornecer os dados:

ID DISCIPLINA: 11764

KEY: 4t8c1B7

- Clicar em "SIM"

- Se apareceu a página da disciplina "Maratona da Semana do IME (2023)", então funcionou!

3. Pode ser consultado qualquer material escrito. Não se pode usar pen-drives nem Internet na competição. Isso desclassifica o participante/equipe.
4. Os problemas serão checados para verificar sua originalidade.
5. Durante a competição as duplas podem solicitar a ajuda da organização apenas para checar se o tratamento de entrada/saída está correto.
6. A classificação final será dada pelo número de problemas corretos com desempate na penalidade. A penalidade é a soma dos tempos de submissão dos problemas certos acrescida de 20 minutos por cada submissão errada.
7. Neste caderno, ao final, há a descrição de um problema simples, de uma maratona anterior da UERJ, também hospedada no site Beecrowd. Ele serve como um exemplo de problema que será dado no dia da competição. Trata-se do problema "Envelopes" (cujo identificador é 3004 no site Beecrowd), que pode ser visto no link

<https://www.beecrowd.com.br/judge/pt/problems/view/3004>

Uma solução a esse problema é apresentada ao final. Neste programa são utilizados os seguintes padrões que devem ser usados nos problemas da competição:

- a) O tratamento de entrada e saída deve ser rígido, exatamente como está na descrição, uma vez que a correção de cada problema é feita de forma automática, comparando a solução gerada pelo programa do aluno com uma solução esperada. Assim, escreva na saída estritamente o que é solicitado, exatamente no formato apresentado. Cuidado com pontuações, sinais de acentuação, quebra de linhas, ou qualquer outro detalhe no formato da saída que pode resultar em detecção de erro por parte da correção automática.
- b) Devido à rigidez da entrada e saída, cuidado para não colocar comandos do tipo CLSCR (que implicitamente envia comandos para a saída), mensagens de

entradas de dados (mensagens do tipo "Entre com o valor de entrada:", quando a descrição da saída não menciona que mensagens como essa devam ser escritas). Lembre-se: o formato da saída deve ser **IDÊNTICO** ao apresentado na descrição.

c) Você deve acreditar que a entrada será exatamente como descrita. Não é necessário acrescentar testes sobre se ela está correta ou não.

d) Outro ponto importante é que haverá vários casos de teste. Para cada caso de teste, você deve imediatamente fornecer a resposta ao problema. Isto é, não acumule entradas para responder mais adiante. O exemplo mostrado para o problema Envelopes ilustra muito bem isso.

e) Se você for testar o problema Envelopes diretamente a partir da compilação, a entrada vai estar misturada com a saída. Não estranhe isso. Além disso, o último caso de teste não será apresentado, já que o programa termina. Se quiser usar um esquema melhor, justamente o utilizado pela correção, crie um arquivo com a entrada (Envelopes.in), gere o executável de seu programa e rode-o no terminal diretamente, usando esse arquivo gerado e endereçando a saída para um arquivo Envelopes.out. Ao abrir esse arquivo, ali constará exatamente a saída produzida pelo programa. É assim que funciona a correção automática do programa.

f) Se estiver programando em C/C++, sempre termine o programa com o comando

```
return 0;
```

9. A descrição de todos os problemas é análoga à do problema Envelopes mostrado neste texto. Sugerimos que você tente resolver os seguintes problemas no site Beecrowd, para se exercitar antes da competição. Assim, você chegará no dia da competição já sabendo tudo sobre o ambiente!

Atenção: não necessariamente os problemas abaixo são simples. Se tiver dificuldades nesses, não se sinta desencorajado de participar da competição!

Aproveite a Oferta:

<https://www.beecrowd.com.br/judge/pt/problems/view/2896>

Pilhas de Paralelepípedos:

<https://www.beecrowd.com.br/judge/pt/problems/view/3005>

O Tasco é sempre Vice:

<https://www.beecrowd.com.br/judge/pt/problems/view/2889>

10. As linguagens permitidas são as do site Beecrowd: praticamente todas as linguagens disponíveis atualmente (inclui C, C++, Java, Javascript, Python)
11. Resolver e programar com calma.

Boa sorte.

Problema Envelopes

Uma empresa quer mandar um envelope para obter uma resposta de um cliente e quer saber se é possível colocar esse envelope dentro de outro. Ambos envelopes são retangulares e um só pode ser colocado dentro do outro se as dimensões forem ambas menores. Dadas as dimensões dos dois envelopes, responda se é possível ou não colocar o primeiro dentro do segundo.

Entrada

A entrada consiste de uma série de testes. A primeira linha contém um único inteiro indicando o número n ($1 \leq n \leq 20$) de casos de testes. A seguir vêm n linhas contendo, cada uma, um caso de teste. Cada caso de teste se compõe de **4 inteiros**: os dois primeiros são as **dimensões** do envelope que deve ir dentro e os dois últimos, **as dimensões** do envelope principal.

Saída

Para cada caso de teste imprima, em uma linha:

- . 'S' se for possível colocar o primeiro envelope dentro do segundo, ou
- . 'N', caso contrário.

Exemplo de Entrada

```
3
10 10 10 20
19 2 3 20
2 20 5 15
```

Exemplo de Saída

```
N
S
N
```

Programa Fonte em C/C++ (Problema Envelopes);

```
#include<stdio.h>
int nc, a, b, c, d, e;
int main() {
    scanf("%ld", &nc);
    for (int i = 1; i<=nc; i++) {
        scanf("%ld %ld %ld %ld", &a, &b, &c, &d);
        if (a > b) {
            e = a; a = b; b = e;
        }
        if (c > d) {
            e = c; c = d; d = e;
        }
        if (a < c && b < d)
            printf("S\n");
        else
            printf("N\n");
    }
    return 0;
}
```

Programa Fonte em Python 3 (Problema Envelopes):

```
nc = int(input())
for i in range(1, nc+1):
    linha = input().split()
    a, b = int(linha[0]), int(linha[1])
    c, d = int(linha[2]), int(linha[3])
    if a > b:
        a, b = b, a
    if c > d:
        c, d = d, c
    if a < c and b < d:
        print("S")
    else:
        print("N")
```