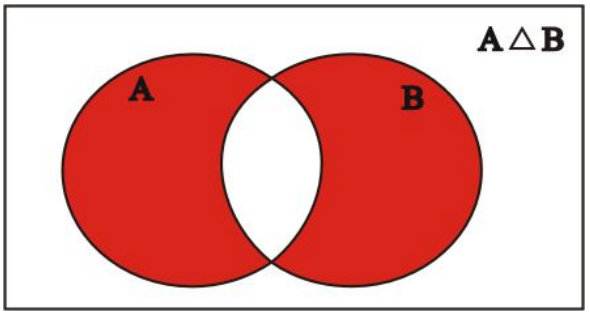
MC202 — ESTRUTURAS DE DADOS

Laboratório 01 — Diferença Simétrica

Paula é uma menina apaixonada por matemática e ela acaba de aprender em sua escola a diferença simétrica de dois conjuntos. A **diferença simétrica** de dois conjuntos é o conjunto de todos elementos que pertencem a algum dos conjuntos, mas não a ambos. Veja o diagrama de Venn que representa a diferença simétrica de $\mathbf{A}$ e $\mathbf{B}$.



Após brincar o dia inteiro com alguns conjuntos, Paula teve a ideia de criar um programa de computador para fazer a diferença simétrica entre dois conjuntos. Porém, ela não sabe programar, então pediu a você para fazer o programa.

Tarefa

A sua tarefa consiste em desenvolver um programa em C que lê dois conjuntos de números inteiros, denotados respectivamente por $\mathbf{A}$ e $\mathbf{B}$, e determine a diferença simétrica desses dois conjuntos, denotada por $\mathbf{S} = \mathbf{A} \Delta \mathbf{B}$.

Como se trata de conjuntos, a ordem dos elementos não é relevante e, portanto, **qualquer permutação** dos elementos de $\mathbf{S}$ será considerada como resposta correta.

Entrada

A entrada é composta por exatamente três linhas. A primeira linha contém dois números inteiros $\boldsymbol{m}$ e $\boldsymbol{n}$ ($1 \leq \boldsymbol{m}, \boldsymbol{n} \leq 40$), que indicam, respectivamente, o número de elementos dos conjuntos $\mathbf{A}$ e $\mathbf{B}$. As duas linhas seguintes contém $\boldsymbol{m}$ e $\boldsymbol{n}$ inteiros que representam os elementos dos conjuntos $\mathbf{A}$ e $\mathbf{B}$, respectivamente.

Saída

O seu programa deve produzir uma única linha de saída correspondente ao conjunto $\mathbf{S}$. O formato da saída apresentada abaixo deve ser seguida rigorosamente, lembrando que qualquer permutação dos elementos será considerada como resposta correta.

Exemplo

Entrada

```
4 5
4 6 7 9
1 2 6 8 9
```

Saída

```
4 7 1 2 8
```

Critérios específicos

- Para as turmas E e F, este laboratório tem peso 1.
- Para as turmas G e H, este laboratório tem peso 1.
- Deverá ser submetido o seguinte arquivo: **lab01.c**.
- Tempo máximo de execução: 1 segundo.

Testando

Nos arquivos auxiliares do SuSy, há um script em Python utilizado para verificar se a solução está correta. Para executá-lo, use o seguinte comando:

```
python3 verifica.py <entrada> <saída>
```

Por exemplo, para compilar com o Makefile fornecido e testar um caso de teste:

```
make
./lab01 < arq01.in > arq01.out
python3 verifica.py arq01.in arq01.out
```

onde `arq01.in` é a entrada (casos de testes disponíveis no SuSy) e `arq01.out` é a saída do seu programa. O Makefile também contém uma regra para testar todos os testes de uma vez; nesse caso, basta digitar:

```
make testar_tudo
```

Observações gerais

No SuSy, haverá 3 tipos de tarefas com siglas diferentes para cada laboratório de programação. Todas possuirão os mesmos casos de teste. As siglas são:

1. **SANDBOX:** Esta tarefa serve para testar o programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém arquivos submetidos aqui **não serão corrigidos**.

2. **ENTREGA:** Esta tarefa tem limite de **uma única submissão** e serve para entregar a versão final dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa: submeta aqui quando não for mais fazer alterações no seu programa.

3. **FORAPRAZO:** Esta tarefa tem limite de **uma única submissão** e serve para entregar a versão final após o prazo estabelecido para o laboratório, mas com nota reduzida (conforme a ementa). O envio nesta tarefa irá substituir a nota obtida na tarefa ENTREGA apenas se o aluno tiver realizado as correções sugeridas no feedback ou caso não tenha enviado anteriormente em ENTREGA.

Observações sobre SuSy:

- Versão do GCC: C-ANSI 4.8.2 20140120 (Red Hat 4.8.2-15).
- Flags de compilação:
  `-ansi -Wall -pedantic-errors -Werror -g -lm`
- Utilize comentários do tipo `/* comentário */`;
  comentários do tipo `//` serão tratados como erros pelo SuSy.

Além das observações acima, esse laboratório será avaliado pelos critérios gerais:

- Indentação de código e outras boas práticas, tais como:
  - uso de comentários (apenas quando forem relevantes);
  - código simples e fácil de entender;
  - sem duplicidade (partes que fazem a mesma coisa).
- Organização do código:
  - tipos de dados criados pelo usuário e funções bem definidas e tão independentes quanto possível.
- Corretude do programa:
  - programa correto e implementado conforme solicitado no enunciado;

- inicialização de variáveis sempre que for necessário;
- dentre outros critérios.