

MC202 — ESTRUTURAS DE DADOS

Laboratório 02 — Bilhetes secretos

Tarefa

Alice e Bob adoram trocar bilhetes secretos na escola. Mas, ocasionalmente, os bilhetes são interceptados e lidos por outras pessoas. Se eles estão dentro da sala de aula, o professor pode ler o bilhete; na hora do recreio, os colegas podem interceptar o bilhete. Então, Alice teve a ideia de trocar dois bilhetes, um na sala de aula e outro na hora do recreio. A ideia é que se apenas um bilhete for interceptado, a mensagem verdadeira não poderá ser lida, mas se a informação dos dois bilhetes for combinada, a mensagem original pode ser lida.

Bob sugeriu que o primeiro bilhete fosse uma mensagem falsa cujas palavras são numeradas começando pelo zero. O segundo bilhete seria uma sequência de índices, cada um correspondendo a uma palavra do primeiro bilhete.

Por exemplo, o primeiro bilhete poderia ser “o rato roeu a roupa do rei de roma” com a seguinte enumeração:

0	1	2	3	4	5	6	7	8
o	rato	roeu	a	roupa	do	rei	de	roma

Se o segundo bilhete fosse a sequência “0 6 2 3 4 5 1”, a mensagem revelada seria “o rei roeu a roupa do rato”.

Seu trabalho será desenvolver um programa para a decifrar a mensagem escondida nos bilhetes.

Entrada

A primeira linha terá no máximo 50000 caracteres e corresponde ao primeiro bilhete (com pelo menos uma e no máximo 1000 palavras). Suponha que cada palavra é formada é uma sequência de uma a 46 letras maiúsculas ou minúsculas sem acentuação.

A segunda linha contém um número inteiro k ($0 \leq k \leq 1000$). A terceira linha contém uma sequência de k índices e corresponde ao segundo bilhete.

Saída

A saída será uma linha contendo a mensagem decifrada.

Exemplo 1

Entrada

```
Vamos concordar que gazear aula e errado
3
0 3 4
```

Saída

```
Vamos gazear aula
```

Exemplo 2

Entrada

```
O professor perdeu no jogo do giz porque ele usa oculos. Aposto
que ele queria ter vencido.
8
0 1 12 9 10 9 6 16
```

Saída

```
O professor que usa oculos usa giz vencido
```

Critérios específicos

- Deverão ser submetidos os seguintes arquivos:
 - lab02.c
- **É obrigatório ler a primeira linha utilizando uma string.**
- Utilize somente a biblioteca `stdio.h`.
- Tempo máximo de execução: 1 segundos.

Dicas

Para ler o primeiro bilhete você precisará de uma função que leia uma linha inteira da entrada padrão, incluindo os espaços. Para isso, podemos usar uma especificação especial de `scanf`.

```
#include <stdio.h>

#define MAX_LINHA 100

int main()
{
    char linha[MAX_LINHA];
    scanf("%[^\n]", linha); // lê caracteres até encontrar \n
    printf("A string lida foi: %s!\n", linha);
}
```

Testando

Para compilar usando o Makefile fornecido e verificar se a solução está correta basta seguir o exemplo abaixo.

```
make
./lab03 < arq01.in > arq01.out
diff arq01.out arq01.res
```

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
```

Isso testará o seu programa com os casos abertos. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados digitando-se:

```
make baixar_fechados
```

Para ver quanto tempo seu programa demora em cada teste, digite:

```
make tempo
```

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. **DRAFT:** 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: gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
- Flags de compilação:
`-std=c99 -Wall -Werror -Werror=vla -pedantic-errors -g -lm`

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

- Indentação de código e outras boas práticas, tais como:
 - uso de comentários (úteis e 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.