MC202 — ESTRUTURAS DE DADOS

Laboratório 05 — Sorteio

Cleber abriu recentemente uma empresa e, com a intenção de atrair mais clientes, decidiu realizar o sorteio de um smartphone. Para participar, cada cliente precisou criar um registro de sorteio contendo seu nome e um número de telefone.

O sorteio funciona da seguinte maneira: os registros são utilizados para montar uma lista em ordem crescente de número de telefone e, no dia da premiação, um número k entre 1 e o número de participantes é sorteado. O cliente premiado será o k-ésimo participante da lista ordenada.

No entanto, há um número extremamente grande de clientes participando. Isto faz com que seja praticamente impossível para Cleber realizar o sorteio conforme proposto sem ajuda de um computador. Então ele resolveu pedir a sua ajuda para desenvolver um programa que recebe a lista de registros e o número k sorteado e determina o cliente premiado.

Tarefa

A sua tarefa consiste em desenvolver um programa em C que lê uma sequência de registros de tamanho indeterminado, o inteiro sorteado e, ao fim, determina o cliente premiado.

Neste laboratório, você deverá criar um registro (struct) que contenha os campos nome e telefone. O seu programa deverá alocar dinamicamente um vetor para manter os registros lidos. Este vetor deve começar com tamanho igual a 4 e dobrar o seu tamanho sempre que estiver cheio. Não é permitido o uso da função `realloc` para alterar o tamanho da memória alocada.

Entrada

A entrada é composta por várias linhas. Cada linha contém uma string que representa o nome de um participante e um inteiro positivo que corresponde ao número de telefone. Cada nome é formado somente por letras minúsculas e sem espaços em branco. Nenhum nome possui mais do que 30 caracteres. A última linha do caso de teste possui o campo nome igual “fim” e um único inteiro k que representa o número sorteado.

Saída

O seu programa deve produzir uma única linha de saída contendo o nome e o número de telefone do cliente premiado.

Exemplo

Entrada

```
joao 98277000
beatriz 33867700
fernanda 35509000
gabriel 85694600
isabelle 38037100
luis 56276843
ana 38249600
sophia 45694400
agatha 81041111
kauan 30847900
gabriela 32790000
vitor 30627700
vitoria 30319655
breno 32980340
rafaela 81069449
nicolash 30319650
tiago 31074999
bruna 88712651
diego 98712750
fim 8
```

Saída

```
beatriz 33867700
```

Testando

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

```
make
./lab05 < 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
```

Critérios específicos

- Para as turmas E e F, este laboratório tem peso 2.
- Para as turmas G e H, este laboratório tem peso 1.
- Deverá ser submetido o seguinte arquivo: **lab05.c**.
- Tempo máximo de execução: 1 segundo.

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.