
Laboratório #2 - lab2.c

Biblioteca 2.0

Prof. Valério Rosset

1 DEFINIÇÃO DO PROBLEMA

A biblioteca do ICT deseja realizar um estudo para coletar o perfil dos estudantes que realizam a reserva de um determinado livro. O objetivo é identificar quantos alunos de cada ano utilizam aquele livro. Considere que a lista de reserva de um determinado livro pode conter no máximo 10 reservas.

TAREFA A

Desenvolva um algoritmo em linguagem C - **utilizando lista estática encadeada** - que forneça uma lista com os números de RA's **ordenados** dos alunos que realizaram a reserva de um determinado livro. Além disso, seu algoritmo também deve fornecer quantos usuários de cada ano realizaram a reserva daquele livro.

Para obter essa informação, considere que o RA dos alunos do ICT muda de acordo com o ano de entrada na universidade. Por exemplo, todos os alunos que entraram em 2013 possuem um RA de 5 dígitos que inicia com 86. Os alunos que entraram em 2014, possuem um RA de 5 dígitos que inicia com 96. Os alunos que entraram em 2015 possuem RA de 6 dígitos que inicia com 106, assim por diante.

ENTRADA

A entrada contém um único conjunto de teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). O conjunto de teste é composto por $L + 1$ linhas, $1 \leq L \leq 100$.

1. A primeira linha contém um valor inteiro L , $1 \leq L \leq 100$, que representa a quantidade de movimentações (reservas e retiradas) que serão realizadas.
2. Em cada uma das L linhas seguintes são dadas as informações. Cada linha contém um valor inteiro - '1' para reservar ou '2' para retirar seguida de um espaço e o número do RA de quem está realizando a movimentação (R), $60000 \leq R \leq 140000$.

Para as operações de entrada, considere a sintaxe:

```
scanf("%d",&variavel);
```

```
scanf("%d %d",&variavel1, &variavel2);
```

SAÍDA

Como saída padrão, seu programa deve imprimir uma lista com os números de RA dos alunos que estão na lista de reserva. Quando um aluno entra na lista de reserva (1) seu número de RA deve ser inserido na lista e quando um aluno sai da lista de reserva (2) seu número de RA deve ser retirado da lista. A lista deve estar ordenada com base nos RA's. Quando ocorrer uma entrada (1) ou uma saída (2) a lista deve ser impressa.

Além disso, as seguintes mensagens também devem aparecer:

1. Tentativa de reserva se a lista estiver cheia: "lista cheia".
2. Tentativa de remoção de um RA que não está na lista: "não existe".

Ao final da execução, você deverá imprimir quantos alunos de cada ano realizaram a reserva daquele livro.

Para as operações de saída, considere a sintaxe:

```
printf("Instrucao de saida\n");
```

```
printf("%d: %d\n", ano, quantidade);
```

```
printf("%d ", itemLista);
```

Entrada	Saída
7	86102
1 86102	66104 86102
1 66104	66104 86102 136105
1 136105	66104 86102 96106 136105
1 96106	86102 96106 136105
2 66104	nao existe
2 66105	86102 96106 136105
1 106100	86102 96106 106100 136105
	2011: 1
	2012: 0
	2013: 1
	2014: 1
	2015: 1
	2016: 0
	2017: 0
	2018: 1

Entrada	Saída
4	86102
1 86102	66104 86102
1 66104	86102
2 66104	
2 86102	2011: 1
	2012: 0
	2013: 1
	2014: 0
	2015: 0
	2016: 0
	2017: 0
	2018: 0

OBSERVAÇÕES

- É fundamental que o programa imprima apenas o valor de saída.
- Os exercícios devem ser postados no site <http://judge.sjc.unifesp.br>.
- Cada aluno deverá postar apenas um exercício dentro da data limite.
- A resolução submetida deve utilizar a estrutura de dados definida no enunciado do exercício.
- Os plágios serão penalizados. Como não será possível saber quem copiou, todos os casos envolvidos receberão punições iguais, portanto, não divulgue seu código.