

Lab SSC0220 – 08/08/2014

Primeiro acesso ao sistema

- 1) Acesse o site SSP: <https://ssp.icmc.usp.br/ssp/site/index2.php>.
- 2) Caso seja o seu primeiro acesso, clique na opção `Cadastro` para realizar o seu cadastro no sistema. Caso contrário, siga para o passo 3.
- 3) Efetue o seu login. Ao entrar na tela principal de sua área, escolha a opção `Matricular` em disciplina, e na tela seguinte, escolha disciplina `Laboratório de ICC 2`. Aperte em `Matricular`. Caso essa disciplina não apareça como uma de suas opções, pode ser que você já esteja matriculado. Volte para a tela principal e verifique em quais disciplinas você está matriculado. Se `Laboratório de ICC 2` for uma delas, parabéns, você foi matriculado com sucesso!

Submetendo um exercício

- 1) Na área `Listar disciplinas matriculadas`, escolha a opção `Submeter exercicio`.
- 2) Na página seguinte, selecione o exercício a ser submetido. Uma janela contendo a descrição do exercício será aberta. Após ler com cuidado o enunciado do exercício/tarefa selecionado, feche a nova janela e volte para a página de submissão.
- 3) Escolha a linguagem de programação `GNU C`, a que nós utilizaremos nesta disciplina.
- 4) Escolha o arquivo com extensão `.c` que você tiver programado previamente para resolver o problema e aperte em `Submeter`.

Pronto! Agora que você já possui todas as instruções necessárias para submeter suas tarefas, vamos começar!!

Na aula de hoje vocês serão incumbidos de fazerem 2 tarefas, sendo a primeira delas uma tarefa com a finalidade de familiarizá-los com o sistema e a segunda dividida em 3 partes as quais serão complementares entre si. As tarefas a serem realizadas serão descritas a seguir, mas antes, fiquem atentos a algumas informações:

Informações

- 1) A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa. Deve-se considerar entradas e saídas padrão;
- 2) Procure resolver o problema de maneira eficiente. As soluções são testadas com outras entradas além das apresentadas nos exemplos;
- 3) Teste seu programa antes de submetê-lo.

Tarefa 1

Problema (retirado de ACM ICPC2008 – South America Regionals/Warm Up):

Og!

Og is a caveman with many children, and he wants to count them all. Og counts his sons with his left hand and his daughters with his right hand.

However, Og is very dumb, and can't add the two counts, so he asked you to write him a program that will do the addition.

Input

The input contains several test cases. Each test case consists of a single line containing two integers L and R, separated by a single space, indicating respectively the number of sons and daughters ($1 \leq L, R \leq 5$).

The end of input is indicated by a line containing only two zeros, separated by a single space.

The input must be read from standard input.

Output

For each test case in the input print a line containing a single integer indicating how many children Og has.

The output must be written to standard output.

Sample input	Output for the sample input
2 2	4
2 3	5
5 5	10
1 1	12
0 0	

Tarefa 2

Busca sequencial

Tarefa 2a – alocação dinâmica: Crie o corpo da função `carrega` descrita a seguir:

```
int* carrega (int N)
```

A função `carrega` deve receber por parâmetro um valor inteiro N que indica a quantidade de posições que devem ser alocadas em um vetor de inteiros. Após fazer a alocação, o valor de cada posição do vetor é lido. O retorno da função é o endereço do vetor alocado.

Tarefa 2a – busca sequencial em vetores: Crie o corpo da função `achou` descrita a seguir:

```
short achou (int vetor[], int N, int v, int *pos)
```

A função `achou` deve receber por parâmetro um vetor de inteiros, seu tamanho, um inteiro v (valor a ser procurado dentro do vetor), e um ponteiro de inteiro que indica a posição em que o valor foi achado. O retorno da função será um `short` indicando se a função achou ou não o valor fornecido.

Tarefa 2c – programa principal: Agora que as tarefas 2a e 2b foram concluídas, chegou a hora de criar o programa principal que chamará as duas funções criadas anteriormente. Lembre-se de desalocar a memória sempre que um vetor não for mais necessário!

Problema:

Crie um programa que leia vários vetores de inteiros e indique quando há ou não um número fornecido como entrada pelo usuário dentro deles.

Entrada

O arquivo de entrada contém diversos casos de teste. A primeira linha de cada caso de teste contém um inteiro N indicando o tamanho do vetor ($1 \leq N \leq 10^4$). A próxima linha contém N inteiros N_i separados por espaços únicos, indicando os elementos existentes no vetor ($1 \leq N_i \leq 10^4$). A linha seguinte contém o valor inteiro V que deverá ser procurado no vetor ($1 \leq V \leq 10^4$). O fim da entrada é indicado por uma linha contendo um 0.

Saída

Para cada caso de teste da entrada, imprima uma linha contendo um único carácter: “S” (letra maiúscula) se o valor for encontrado, e “N” (letra maiúscula) caso contrário.

Exemplo de entrada	Saída para o exemplo de entrada
3 500 700 600 100 9 200 51 6 102 10 400 150 9 83 5 5 3 4 5 6 1 4 0	N N S