



OBI2009

Caderno de Tarefas

Modalidade **Programação** • Nível **2**, Seletiva IOI, Dia 1

A PROVA TEM DURAÇÃO DE 2 HORAS

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 2 páginas (não contando esta folha de rosto), numeradas de 1 a 2. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em Pascal: *readln*, *read*, *writeln*, *write*;
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
- Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Promoção:

Sociedade Brasileira de Computação

Patrocínio:

Fundação Carlos Chagas

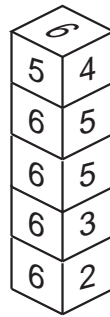
Pilha de Dados

Nome do arquivo fonte: dados.c, dados.cpp, ou dados.pas

Zeinho tem um conjunto de dados e gosta muito de brincar de fazer pilhas de dados. Cada dado de Zeinho é um cubo perfeito; cada uma das seis faces de um dado tem um número inteiro, entre um e seis. Os dados não são dados normais, ou seja, a soma dos números em faces oposta não é necessariamente sete, mas cada número entre um e seis aparece em exatamente uma face de cada dado.

Zeinho montou uma pilha de dados e quer girar os dados, sem mudar a ordem deles, de forma que:

- para dois dados adjacentes na pilha, as faces que se tocam tem o mesmo número;
- a soma dos números de uma das faces da pilha deve ter o maior valor possível.



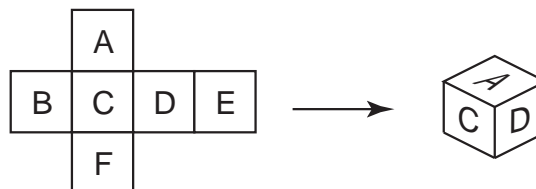
Tarefa

Sua tarefa é escrever um programa que, dada a descrição da pilha de dados, calcula o valor máximo da soma dos números em uma das faces da pilha.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado).

A primeira linha da entrada contém um número inteiro N , indicando o número de dados na pilha ($1 \leq N \leq 10^4$). Cada uma das N linhas seguintes descreve um dado. A i -ésima linha contém seis números inteiros entre 1 e 6, separados por um espaço em branco, indicando os números nas faces do i -ésimo dado. Os números são dados na ordem A, B, C, D, E e F, correspondendo às faces indicadas na figura abaixo.



Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha contendo o maior valor possível que pode ser obtido na soma dos números de uma das faces da pilha girando-se os seus dados.

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 25 pontos, $N \leq 10$.

Exemplos

Entrada	Saída
2 1 2 3 4 5 6 1 2 3 4 5 6	12

Entrada	Saída
5 2 3 1 6 5 4 3 1 2 4 6 5 5 6 4 1 3 2 1 3 6 2 4 5 4 1 6 5 2 3	29