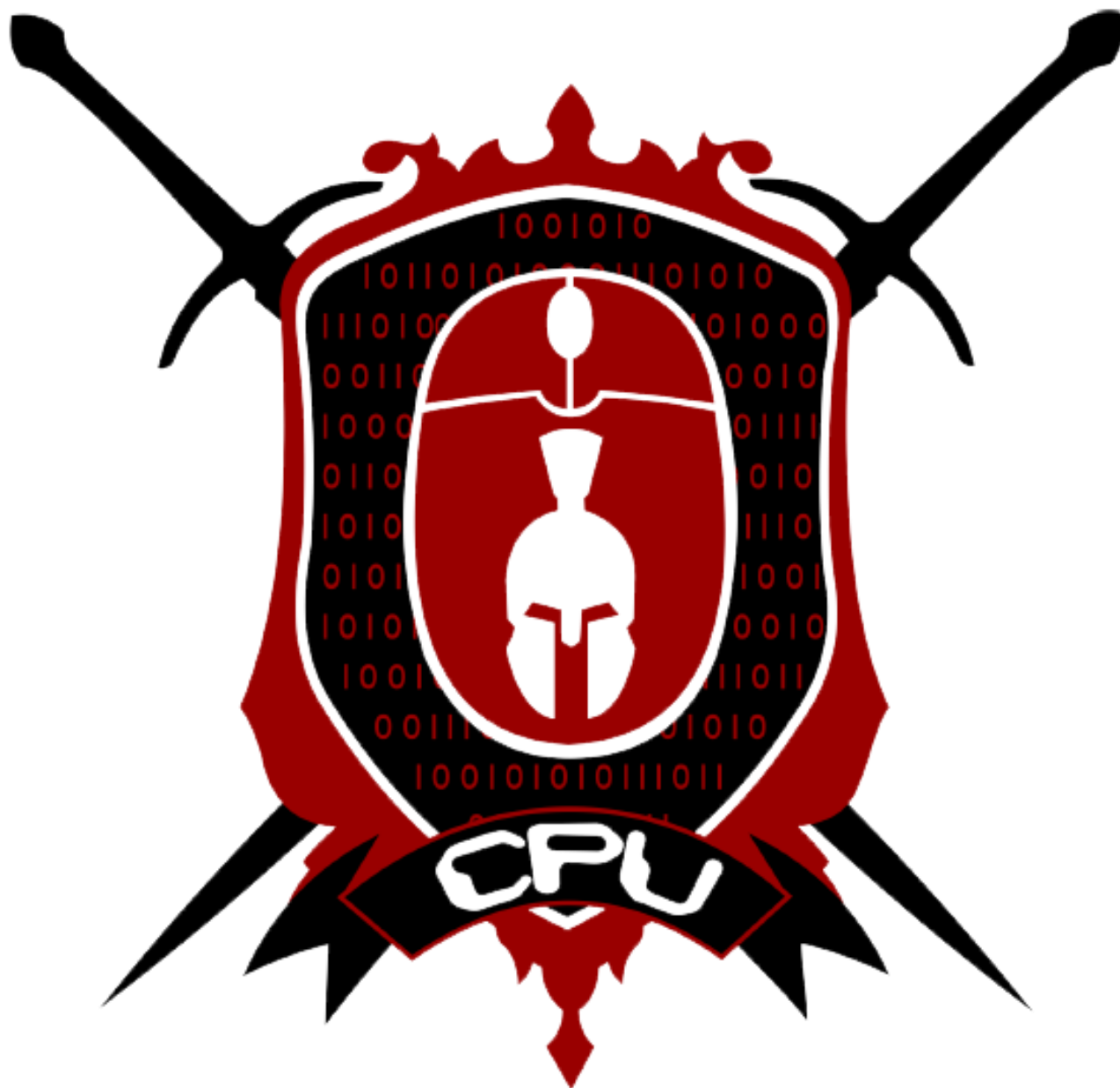


## CADERNO DE QUESTÕES



## Questão 1) A Família

Arquivo: familia.[c|cpp|java]

Família é uma união de pessoas com vínculo sanguíneo, isto é, advindas de uma mesma linhagem. Em uma família, as pessoas se ligam por graus de parentesco, pessoas com o pai em comum são irmãos, pessoas com avós em comum e pais diferentes são primos, e assim vai.

João, um dia, fez uma árvore genealógica da sua família, começando pelo seu tataravô (pai de seu bisavô) e listou seus descendentes desde então. Segue abaixo a descrição da árvore de João.

Astolfo é pai de José Augusto e de Ana Maria; José Augusto deu origem à Pedro e Luiz; já Ana Maria só teve sua primogênita, Luiza; Pedro, por sua vez, gerou Jefferson; Luiz gerou João Augusto e Jenilson; Luiza deu à luz Carla e Marina; João Augusto então gerou João e Alberto; por fim, Marina gerou Josefina e Carlota.

### TAREFA

Conhecendo a árvore genealógica de João (descrita acima) faça um programa que determine o grau de parentesco entre dois membros desta família.

### ENTRADA

A entrada terá vários casos de teste. Cada caso teste consiste em duas linhas cada uma contendo os dois membros da família de João, na primeira linha tem-se A e na segunda tem-se B. Os membros da família de João serão identificados pelo nome (você pode desconsiderar a acentuação).

O fim da entrada é dado quando A e B são idênticos.

### SAÍDA

Para cada caso de teste deverá ser impressa em uma única linha o grau de parentesco de A em relação à B. Os graus de parentesco poderão ser: tataravô, bisavô e bisavó, avô e avó, pai e mãe, irmão e irmã, tio e tia, tio-avô e tia-avó, sobrinho e sobrinha, filho e filha, neto e neta, bisneto e bisneta, tataraneto e tataraneta, e primo e prima (para qualquer grau de proximidade entre os primos)

Você poderá desconsiderar acentuações, mas atentar-se para o gênero dos familiares.

### EXEMPLO

Entrada:

```
Joao
Jefferson
Luiza
Carla
Astolfo
Marina
Joao
Joao
```

Saída:

Primo  
Mae  
Bisavo

## Questão 2) Calendário Lunar

Arquivo: calendario.[c|cpp|java]

Antes que os Portugueses chegassem ao Brasil nativo, introduzindo à cultura local a contagem do tempo em dias, meses e anos, é afirmado por alguns historiadores que os habitantes nativos da Nova Terra utilizassem as fases da lua para saber quanto tempo havia passado. Por observação, os nativos sabiam que após a Lua Cheia viria a Lua Minguante e então a Lua Nova e, por fim, a Lua Crescente; e após esta, tem-se a Lua Cheia novamente. Eles também sabiam que a Lua Cheia pairava nos céus durante o mesmo montante de tempo que as demais (em dias este montante equivale a 7).

### TAREFA

Sabendo que o dia 1/1/1970 correspondeu a um primeiro dia de Lua Cheia, faça um programa que receba uma data no formato DD/MM/AAAA e informe qual lua estará pairando sob os céus neste dia.

### ENTRADA

Cada entrada é composta por vários casos de teste. Cada caso de teste estará em uma linha distinta da entrada e apresentará um conjunto de 3 (três) inteiros positivos D, M e A, todos separados entre si por uma barra normal. D corresponde ao dia data de teste ( $01 \leq D \leq 31$ ), M representa o mês da data de teste ( $01 \leq M \leq 12$ ) e A representa o ano da data de teste ( $1970 \leq A \leq 3000$ ).

A entrada é encerrada quando D for igual à zero (0).

Você poderá considerar que:

- A data sempre será válida quanto à quantidade de dias em relação ao mês e ao ano;
- O ano bissexto mais próximo de 1970 é 1972.

### SAÍDA

Para cada caso de teste deve ser impressa uma linha distinta informando a data do caso de teste no formato DD/MM/AAAA e a lua que estará no céu neste dia, separados por um espaço em branco. O nome da lua deve estar por extenso e em letras maiúsculas.

### EXEMPLO

Entrada:

26/09/2012  
15/04/1977  
31/12/2888  
00/12/2012

Saída:

26/09/2012 MINGUANTE  
15/04/1977 CHEIA  
23/12/2888 NOVA

### Questão 3) O Display Enigmático

Arquivo: display.[c|cpp|java]

O banco no qual você é programador recentemente instalou um caixa eletrônico em uma região onde existem muitos ‘trombadinhas’ (ladrões ocasionais). Preocupada com a segurança da senha de seus clientes, foi pedido que você desenvolvesse um novo display para a digitação da senha no terminal de atendimento.

Foi estabelecido que cada vez que a senha fosse solicitada ao cliente, este tivesse que digitá-la utilizando apenas 5 letras aleatórias e distintas entre si e que, para cada uma destas letras exibidos na tela, um par de números naturais aleatórios fosse associado a ele.

#### TAREFA

Sua tarefa é fazer um programa que dada uma quantidade  $N$  de combinações possíveis para a sequência digitada por um cliente, sob  $N$  condições de associação, para a mesma senha retorne a senha do cliente em questão.

#### ENTRADA

Cada entrada é composta por vários casos de teste. Um caso de teste é iniciado por um inteiro positivo  $N$  ( $2 \leq N \leq 10$ ) que indica quantas combinações serão necessárias para descobrir a senha, seguindo-se por  $N$  linhas contendo 5 letras e 10 números naturais, em ordem de associação respectiva às letras (o primeiro par de números é associado à primeira letra, o segundo par à segunda letra, e assim por diante), e 6 letras correspondentes à combinação digitada. O fim da entrada é indicado por  $N = 0$ .

#### SAÍDA

Para cada caso de teste deve ser impressa uma linha distinta informando a senha digitada sob as referentes condições de associação onde cada número formador da senha deve estar separado por um espaço em branco.

#### EXEMPLO

Entrada:

```
2
a e i o u 2 3 4 5 6 7 8 9 0 1 a a a u o i
f h q y u 1 5 2 6 0 7 3 9 4 8 h y y f u q
3
a b c d e 0 1 2 3 4 5 6 7 8 9 b c d d e e
```

```
a b c d e 1 3 5 4 6 8 7 9 0 2 e b c d c d
a b c d e 2 1 0 4 5 9 7 6 8 3 a c d d e c
0
```

Saída:

```
2 3 3 1 8 7
2 5 6 7 8 9
```

#### Questão 4) Segundo Turno

Arquivo: votacao.[c|cpp|java]

Uma escola está elaborando uma votação para decidir qual o aluno mais popular. Como o voto é secreto, os organizadores decidiram que o eleitor poderia votar em qualquer número de candidatos (marcar qualquer número de células de candidatos na cédula de votação) e poderá, também, votar em si próprio. Porém, percebeu-se que o resultado final não poderia indicar com clareza significativa qual o aluno escolhido para o posto de mais popular. Por fim, foi decidido que caso o montante de votos do primeiro candidato não atingisse 45% de representatividade na eleição, este deveria integrar uma nova votação com o segundo colocado, estabelecendo o famoso segundo turno.

#### TAREFA

Você deve fazer um programa que deverá computar todos as cédulas, computando os votos aos devidos candidatos, e indicar qual o aluno escolhido ou se deverá ser realizado, e entre quem, o segundo turno.

#### ENTRADA

Uma entrada apresenta um número não determinado de casos de teste. A primeira linha de um caso de teste possui um inteiro positivo  $N$  ( $2 \leq N \leq 80$ ) que indica quantos alunos existem na escola; as próximas linhas do caso de teste correspondem, cada uma, a uma cédula de votação na qual cada inteiro  $V$ , separado pelo próximo inteiro por um espaço em branco, pode ser zero (não computar voto) ou um (computar um voto). Note que o  $n$ -ésimo inteiro  $V$  de uma linha indica se o  $n$ -ésimo candidato recebeu ou não o voto.

O fim da entrada é sinalizado por  $N = 0$ .

#### SAÍDA

Para cada caso de teste deve ser impressa uma linha dizendo: “o <aluno> e o mais popular” caso haja um vencedor, ou “2 turno entre <aluno1> e <aluno2>”, caso haja a necessidade de segundo turno.

Note que <aluno>, <aluno1> e <aluno2> correspondem ao identificador ordinal do candidato (1candidato, 2candidato, etc.) e <aluno1> e <aluno2> correspondem, respectivamente, ao primeiro colocado e ao segundo colocado na eleição.

Em caso de empate entre candidatos, deve-se considerar o candidato de menor identificação ordinal.

### EXEMPLO

Entrada:

```
5
1 0 0 1 0
0 0 1 0 0
0 1 1 0 0
1 1 0 1 0
1 0 0 0 0
3
1 1 0
0 1 0
1 1 1
0
```

Saída:

```
2 turno entre 1candidato e 2candidato
o 2candidato e o mais popular
```

### Questão 5) Engrenagens

Arquivo: engrenagens.[c|cpp|java]

As empresas Kiold trabalham há quase 100 anos na produção de peças para carros de luxo.

Apesar de ser uma empresa de grande porte, algumas de suas máquinas são muito antigas, exigindo peças antigas e que devem ser substituídas de tempos em tempos. Uma dessas máquinas é uma estofadora de 1985 que é usada para a costura de dados de pelúcia. O tamanho do dado é definido pela quantidade de engrenagens colocadas na máquina, fazendo assim, com que ela possa ter várias configurações diferentes. Ela opera com engrenagens encaixadas em uma barra cilíndrica que atravessa os eixos centrais de cada engrenagem, fazendo com que todas girem à mesma velocidade angular.

Essas engrenagens tem vida útil em ciclos (giros completos) e cada vez que uma engrenagem falha a máquina pára e a engrenagem defasada é trocada por outra com as mesmas propriedades, porém nova.

### TAREFA

Dados os ciclos máximos (vida útil) de cada engrenagem antes que ela seja substituída, você deverá fazer um programa que diga em quantos ciclos completos da primeira engrenagem todas as engrenagens falharão de uma só vez pela primeira vez.

### ENTRADA

A entrada possui inúmeros casos de teste. Um caso de teste é iniciado por uma linha contendo apenas um inteiro positivo  $E$  ( $2 \leq E \leq 20$ ) que indica quantas engrenagens integram o sistema da

máquina e termina com uma única linha contendo E inteiros positivos C ( $1 \leq C \leq 1000$ ), separados entre si por um espaço em branco, que indicam a vida útil de cada engrenagem do sistema; note que o primeiro inteiro C de uma linha representa a vida útil da primeira engrenagem, o segundo inteiro C representa a vida útil da segunda engrenagem, e assim por diante.

O fim da entrada é sinalizado por  $E = 0$ .

### SAÍDA

Para cada caso de teste deve-se exibir em uma única linha um inteiro representando o número de ciclos que a primeira engrenagem efetuou até a primeira falha total do sistema de engrenagens.

### EXEMPLO

Entrada:

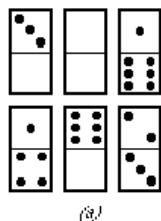
```
3
3 2 1
3
5 2 2
0
```

Saída:

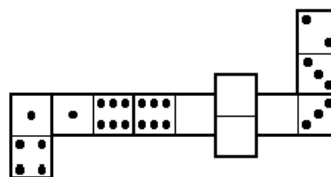
```
6
10
```

**Questão 6) Dominós** (Retirado de <http://olimpiada.ic.unicamp.br/pratique/programacao/nivel2/domino>)  
Arquivo: dominos.[c|cpp|java]

Todos conhecem o jogo de dominós, em que peças com dois valores devem ser colocadas na mesa em sequência, de tal forma que os valores de peças imediatamente vizinhas sejam iguais. O objetivo desta tarefa é determinar se é possível colocar todas as peças de um conjunto dado em uma formação válida.



(a)



(b)

Conjunto de seis peças (a) e uma formação utilizando todas as seis peças (b)

### TAREFA

É dado um conjunto de peças de dominó. Cada peça tem dois valores X e Y, com X e Y variando de 0 a 6 (X pode ser igual a Y). Sua tarefa é escrever um programa que determine se é possível organizar todas as peças recebidas em sequência, obedecendo às regras do jogo de dominó.

### ENTRADA

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de testes contém um número inteiro  $N$  ( $2 \leq N \leq 100$ ) que indica a quantidade de peças do conjunto. As  $N$  linhas seguintes contêm, cada uma, a descrição de uma peça. Uma peça é descrita por dois inteiros  $X$  e  $Y$  ( $0 \leq X \leq 6$  e  $0 \leq Y \leq 6$ ) que representam os valores de cada lado da peça.

O final da entrada é indicado por  $N = 0$ .

## SAÍDA

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste  $n$ ", onde  $n$  é numerado a partir de 1. A segunda linha deve conter a expressão "sim" se for possível organizar todas as peças em uma formação válida ou a expressão "nao" (note a ausência de acento) caso contrário. A terceira linha deve ser deixada em branco.

## EXEMPLO

Entrada:

```
3
0 1
2 1
1 1
2
1 1
0 0
6
3 0
0 0
1 6
4 1
0 6
2 3
0
```

Saída:

```
Teste 1
sim

Teste 2
nao

Teste 3
sim
```

**Questão 7) Algoritmos e Eficiência** (Retirado de <http://online-judge.uva.es/p/v1/100.html>)

Arquivo: algoritmos.[c|cpp|java]

Problemas em ciência da computação são classificados como pertencentes a uma determinada classe de problemas (NP, sem solução, recursivos). Nesse problema você irá analisar o seguinte algoritmo:

1. entrada  $n$



2. mostre  $n$
3. se  $n = 1$  então STOP
4. se  $n$  é ímpar então
5.        $n \leftarrow 3n+1$
6. senão
7.        $n \leftarrow n/2$
8. GOTO 2

Dada a entrada 22, a sequência de números que será mostrada é 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.

É conjecturado que o algoritmo acima terminará (quando 1 for mostrado) para qualquer valor inteiro de entrada. Apesar da simplicidade do algoritmo, é desconhecido que essa conjectura é verdadeira. Isso foi verificado, entretanto, para todos os inteiros  $n$  tais que  $0 < n < 1000000$ .

Dada uma entrada  $n$ , é possível determinar o número de números mostrados (incluindo o 1). Para um dado  $n$ , isso é chamado de comprimento do ciclo de  $n$ . No exemplo acima, o comprimento do ciclo de 22 é 16.

#### TAREFA

Nesta questão você deverá fazer um programa que, para quaisquer dois números, determine o maior comprimento de ciclo entre estes números (considerando a análise do algoritmo exibido acima).

#### ENTRADA

A entrada consistirá de uma série de pares de inteiros  $i$  e  $j$ , um par de inteiro por linha. Todos os inteiros serão menores que 1000000 e maiores que 0.

Você deverá processar todos os pares de inteiros e para cada par, determinar o máximo comprimento de ciclo entre todos os inteiros, incluindo o  $i$  e  $j$ .

Você pode assumir que nenhuma operação excede o tamanho de um inteiro de 32 bits.

A entrada é finalizada com  $i$  ou  $j$  igual à zero.

#### SAÍDA

Para cada par de inteiros da entrada, você deverá mostrar a saída  $i$ ,  $j$  e o ciclo de comprimento máximo encontrado. Esses três números deverão ser separados entre si por um espaço em branco.

#### EXEMPLO

Entrada:

```
1 10
100 200
201 210
900 1000
0 0
```

Saída:

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

### Questão 8) Lançando Foguetes

Arquivo: foguete.[c|cpp|java]

João é um projetista de foguetes. Durante muitos anos, João viu as suas criações ganharem os céus. Porém, o ato de ficar muito tempo com o pescoço inclinado, trouxe um desconforto para João, que já não pode apreciar com naturalidade os seus foguetes, devido às dores no pescoço.

#### TAREFA

Desenvolva um programa que, sabendo as circunstâncias do ambiente, informe quanto tempo após o lançamento do foguete João pode manter a cabeça inclinada acompanhando a trajetória do foguete sem sentir dores.

#### ENTRADA

A entrada consiste em vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha contendo quatro números reais positivos separados por um espaço entre si: D, L, V e A; onde D ( $10 \leq D \leq 10000$ ) é dado em metros e simboliza a distância entre João e seu foguete, L ( $5 \leq L \leq 45$ ) é o comprimento, também em metros, do foguete, V ( $20 \leq V \leq 720$ ) é a velocidade de ascensão, dada em metros por segundo, do foguete e A ( $1 \leq A < 90$ ) é o ângulo máximo, dado em graus que João pode inclinar a cabeça sem sentir dores.

A entrada termina quando todos os valores são nulos (iguais a zero).

#### SAÍDA

Para cada conjunto de teste deve ser exibido em uma única linha um número real com precisão de duas casas decimais que indique o tempo que João pode observar seu foguete sob as referentes circunstâncias.

#### EXEMPLO

Entrada:

```
800 30 32 30
1000 52 72 30
0 0 0 0
```

Saída:

```
13.48
7.29
```

### Questão 9) Verificador de Trapaças

Arquivo: trapaca.[c|cpp|java]

José e Antonio estão jogando um jogo de adivinhação onde o objetivo é descobrir, dentro de uma sequência limitada, o número que o outro jogador está pensando. Para facilitar, quem estiver pensando no número deve informar a cada palpite do advinho se o palpite está alto demais, baixo demais ou se está correto.

Apesar de ambos serem muito amigos, Antonio costuma trapacear respondendo erroneamente aos palpites de José.

### TAREFA

Você deverá desenvolver um programa que computará os palpites de José e as respostas de Antonio e retornará se Antonio está trapaceando ou não assim que uma ameaça de trapaça for encontrada.

### ENTRADA

A entrada é composta de vários casos de teste. Cada caso começa com dois inteiros A e B ( $A > B$  e  $B > 0$ ), separados por um espaço em branco, que correspondem ao valor máximo e mínimo possível no jogo, após isto, em outra linha, segue-se o primeiro palpite de José e a partir disto alterna-se entre a resposta de Antonio e o novo palpite de José, todos sempre em uma nova linha.

Os palpites de José são sempre inteiros positivos pertencentes ao intervalo formado por A e B, já as respostas de Antonio podem ser “ALTO DEMAIS”, “BAIXO DEMAIS” ou “CORRETO”, indicando a relação do palpite de José com o número pensado.

A entrada é terminada quando A e B são iguais. Os casos de teste, por sua vez, terminam quando a resposta de Antonio é “CORRETO”.

### SAÍDA

A saída do seu programa deve ser: “Antonio joga limpo” caso nenhuma ameaça de trapaça seja encontrada, ou “Antonio joga sujo” caso seja notada uma resposta trapaceira de Antonio.

### EXEMPLO

Entrada:

```
10 2
8
ALTO DEMAIS
5
BAIXO DEMAIS
7
BAIXO DEMAIS
8
CORRETO
6 10
8
BAIXO DEMAIS
10
ALTO DEMAIS
9
CORRETO
```

5 5

Saída:

```
Antonio joga sujo  
Antonio joga limpo
```

### Questão 10) Lançamento de Dados

Arquivo: dados.[c|cpp|java]

Um dos instrumentos geradores de resultados aleatórios mais eficaz e conhecido é o dado de seis faces, uma peça de formato cúbico onde cada face tem um número inteiro associado a cada face começando pelo 1 (um).

O resultado do seu funcionamento é imprevisível, entretanto, por se tratar de um instrumento com dimensões limitadas, pode-se calcular a probabilidade da ocorrência de um resultado. Em uma jogada com apenas um dado é possível saber a probabilidade de um evento obtendo o quociente entre a quantidade de vezes que a condição do evento está presente no domínio e o próprio domínio. Mas quando se lançam combinações (somadas) de dados o raciocínio é outro, podendo, inclusive, gerar possibilidades distintas para as ocorrências de diferentes números.

#### TAREFA

Desenvolva um programa que sob o lançamento de N dados de seis faces informe quais números terão mais chance de serem obtidos através da soma dos resultados individuais de cada dado.

#### ENTRADA

A entrada é composta de vários casos de teste. Cada caso é composto por um número natural N ( $2 \leq N \leq 5$ ).

A entrada se encerra quando N é igual a 0 (zero).

#### SAÍDA

A saída do seu programa deve mostrar, para cada caso de teste, em apenas uma linha, o número de dados lançados e, em seguida, os números cuja probabilidade de ocorrência seja a maior entre os demais.

#### EXEMPLO

Entrada:

```
2  
5  
0
```

Saída:

```
2 7  
5 15
```