



MARATONA DE PROGRAMAÇÃO 2019-1

MARATONA
ROBÓTICA PAULA SOUZA 
 2019-1

CADERNO DE PROBLEMAS



São Paulo – Brasil

Junho, 2019

**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2019-1**

Instruções

- 1) Este caderno contém 6 problemas (A-F). As páginas estão numeradas de 1 a 11, não contando a página de rosto. Verifique se o caderno está completo antes de começar;
- 2) Em todos os problemas, os programas deverão ler a entrada padrão para recebimento de dados externos e mostrar na saída padrão os resultados esperados;
- 3) Em todos os problemas, em que o final da entrada não esteja especificado no texto do problema, deverá ser considerado o final das entradas;
- 4) Não é permitido em nenhuma hipótese a utilização de código pronto. Os códigos poderão ser consultados de anotações e publicações em geral, desde que escritas ou impressas, e deverão ser digitados na hora da competição;
- 5) É terminantemente proibido a utilização da Internet para buscar códigos prontos ou qualquer outra informação, que não seja acesso e utilização da plataforma oficial da competição, isto é, o software BOCA;
- 6) Também não será permitido o uso de pendrives, hds externos, telefones celulares, tablets, ou qualquer outro dispositivo eletrônico, que não o único computador disponibilizado ao time;
- 7) Qualquer desrespeito às normas de restrições acima, o time será desclassificado;
- 8) Qualquer dúvida contate o pessoal do staff.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Dicas para Iniciantes

Aqui vão algumas dicas rápidas para quem não está acostumado com os tipos de problemas da maratona e de olimpíadas de informática em geral:

- Todo problema tem uma entrada exemplo e a saída esperada para aquela entrada. Você deve sempre se lembrar que essa entrada não é nem de perto a entrada que os juízes irão usar para testar seu programa. Lembre-se sempre de que é apenas um exemplo. Após fazer o programa funcionar para os exemplos, teste sempre condições extremas e de contorno (verifique os limites do problema, qual os valores mínimo e máximo que as variáveis podem atingir, etc.);
- Evite mandar programas precipitadamente. É bem melhor gastar 5 minutos testando o programa para ter certeza de que funciona para as mais diversas entradas do que receber uma mensagem de erro dos juízes (que acarreta uma penalização de 10 minutos);
- A maioria dos problemas sempre tem um "truque" escondido. Por exemplo, suponha um problema trivial de calcular fatorial. Você se lembrou de tratar o problema quando a entrada é zero? Sempre procure pensar no que o juiz deveria estar pensando para fazer a entrada. Afinal, o seu programa deve funcionar corretamente para qualquer entrada;
- Se o seu algoritmo lidar com busca, procure cortar ao máximo. Quanto menos nós expandidos, mais eficiente seu programa. Problemas de busca são, em geral, críticos com relação ao tempo, e um descuido pode acarretar em estouro do tempo limite!;
- A maratona é um torneio de criação e implementação de algoritmos. Isto significa que você nunca vai precisar se preocupar com coisas do tipo interface com o usuário, reusabilidade do código, etc. Aliás, é bem melhor usar nomes sugestivos de variáveis do que comentários;
- Evite usar as ferramentas de debug, elas consomem muito tempo. Aliás, deve-se evitar "debugar" o problema no computador, afinal, são três pessoas para apenas um computador! Enquanto você estiver "debugando" o programa na tela do computador, outros componentes do time podem estar querendo digitar um programa! O ideal é imprimir o código fonte (isso é permitido na maratona) e tentar analisar o código em busca de erros. Se for indispensável o "debug" em "tempo real" procure fazê-lo usando "print" de variáveis e técnicas do tipo. Às vezes breakpoints também podem ser úteis;
- Ninguém vai analisar seu código, portanto não interessa se ele está elegante ou não. Se você tiver uma solução "feia", porém eficiente, essa é a que se deve usar;
- Nem sempre um algoritmo polinomial pode ser viável. Suponha que você tenha implementado um algoritmo $O(n^3)$ para um determinado problema. Mas e se n puder assumir valores até, digamos, 1000? Com quase toda certeza seu problema irá estourar o limite de tempo. Por isso, sempre preste atenção não só à complexidade do seu algoritmo, como à viabilidade de sua implementação;
- Não há nada pior do que gastar muito tempo fazendo e implementando um algoritmo para, depois, descobrir que ele está errado. Numa maratona, esse erro é fatal. Por isso, apesar de a pressa ser necessária, procure sempre certificar-se da correção do algoritmo ANTES de implementá-lo!

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema A

Latitude e Longitude

Arquivo fonte: `LatLong.{ c | cc | java | py2 | py3 }`
Autores: Prof. Hamilton Machiti (ETEC São Bernardo do Campo) e
Prof. Henrique Louro (ETEC de Caraguatatuba)

Tarefa

Latitude e longitude são descrições da localização, ou coordenadas geográficas, de um determinado lugar na Terra. O modo como a latitude é definida depende da superfície de referência utilizada, e a longitude é medida em graus, de zero a 180 para leste ou para oeste, a partir do Meridiano de Greenwich, porém não há uma posição inicial natural para marcar a longitude.

Latitude é o ângulo entre o plano do equador à superfície de referência. A latitude mede-se para norte e para sul do equador, entre 90° sul, no Polo Sul e 90° norte, no Polo Norte. A latitude é a distância ao Equador medida ao longo do meridiano de Greenwich, esta distância mede-se em graus, podendo variar entre 0° (no equador) e 90° para Norte ou 0° (no equador) e -90° para Sul.

Por outro lado, a **longitude** é medida ao longo do Equador, e representa a distância entre um ponto e o Meridiano de Greenwich. Também é medida em graus, podendo ir de 0° a 180° para Leste ou de 0° a -180° para Oeste, conforme Figura 1 abaixo.

Além disso, tanto a **latitude** quanto a **longitude**, tem suas subdivisões em minutos e esses em segundos, onde: 1° = 60'(minutos), 1' = 60"(segundos).

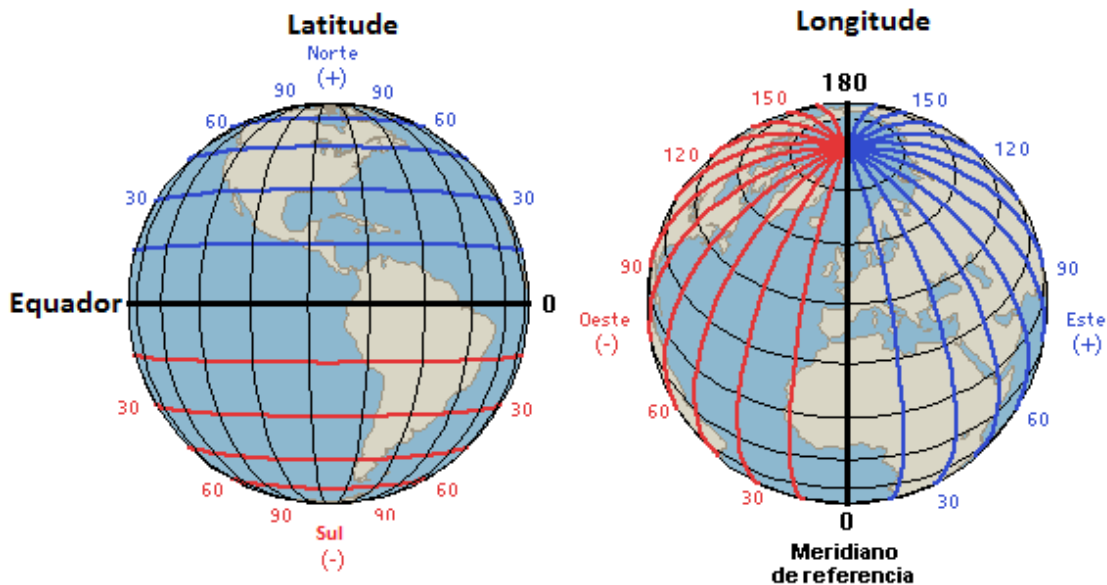


Figura 1 - Ilustração de Latitude e Longitude

Fonte: <http://forest-gis.com/2011/07/como-converter-graus-minutos-e-segundos-para-graus-decimais.html/>

A medição da longitude é importante tanto para a cartografia como para uma navegação segura no oceano. Ao longo da história, muitos exploradores lutaram para encontrar um método de determinar a longitude, como Américo Vespúcio e Galileu. Porém, o cálculo da longitude sempre apresentou sérios problemas, principalmente no alto mar. Determinar a latitude é mais simples, basta medir o ângulo entre o horizonte e a Estrela Polar com ajuda de um quadrante, astrolábio ou sextante.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Exemplo

A nossa posição sobre a Terra é referenciada em relação ao equador e ao meridiano de Greenwich e baseia-se em três denominações: **a latitude, a longitude e a altitude.**

A latitude e longitude de Brasília (a capital brasileira) é 15° 46' 47" Sul e 47° 55' 47" Oeste, ou -15° 46' 47" e -47° 55' 47", ou -15,7796° e -47,9296°..

Para converter minutos e segundos para graus, veja o exemplo:

$$-15^{\circ} 46' 47''$$

$$-15 + -46/60 + -47/3600 = -15 + -0,7666 + -0,0130 = -15,7796^{\circ}$$

$$-47^{\circ} 55' 47''$$

$$-47 + -55/60 + -47/3600 = -47 + -0,9166 + -0,0130 = -47,9296^{\circ}$$

Sua tarefa é criar um programa, que dado dois números com casas decimais, um pertencente à latitude, e outro pertencente à longitude, mostre na saída padrão qual é a sua posição, mediante aos pontos cardeais: (Norte, Sul, Leste ou Oeste). Por exemplo: latitude de -32° e longitude de 85°, resulta na saída "Sul Leste".

Entrada

O arquivo de entrada terá vários casos de teste. Cada caso, contido em uma linha, será formado por dois números reais, latitude (LT) e longitude (LG), sendo $-90^{\circ} \leq LT \leq 90^{\circ}$ e $-180^{\circ} \leq LG \leq 180^{\circ}$, separados por um ponto-e-vírgula. As entradas deverão ser lidas da entrada padrão. Uma linha com apenas um número 0 encerra as entradas.

Saída

Para cada caso de teste, seu programa deverá mostrar uma linha com duas palavras, informando o ponto cardinal correspondente (Norte, Sul, Leste, Oeste, Equador e Greenwich). As saídas deverão ser escritas na saída padrão.

Exemplo de entrada	Saída para o exemplo de entrada
-23° 0" 00.00';-46° 0" 00.00'	Sul Oeste
37° 5' 24.72";-95° 42' 46.44"	Norte Oeste
0° 0" 00.00',-10° 0" 00.00'	Equador Oeste
20° 0" 00.00';0° 0" 00.00'	Norte Greenwich
0	

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema B Cidade Sortuda

Arquivo fonte: Cidade.{ c | cc | java | py2 | py3 }
Autores: Prof. Hamilton Machiti (ETEC São Bernardo do Campo) e
Prof. Henrique Louro (ETEC de Caraguatatuba)

Tarefa

A Mega-Sena é uma das loterias administradas pela Caixa Econômica Federal no Brasil. Os sorteios são realizados na localidade programada para o Caminhão da Sorte, em locais de grande evidência pública, previamente escolhidos e divulgados (Sorteio Itinerante), no Espaço CAIXA Loterias no Terminal Rodoviário Tietê, em São Paulo/SP e no Auditório da CAIXA, em Brasília/DF.

São sorteadas 6 dezenas por concurso. Os jogadores podem ganhar prêmios se acertarem 6 dezenas (Sena), 5 dezenas (Quina) e 4 dezenas (Quadra). Se houver empate em qualquer uma delas, o prêmio é rateado pelo número de ganhadores.

A partir de maio/2009 a CEF passou a informar as cidades e os estados brasileiros de onde são os sortudos ganhadores da Sena. Assim, é possível saber qual as cidades brasileiras com o maior número de ganhadores, até hoje, e os prêmios que foram pagos.

Além disso, a caixa disponibiliza um arquivo com todas as informações dos concursos, desde o início dessa loteria.

Sua tarefa será a de ler da entrada padrão alguns dados descritos abaixo e calcular quais as três cidades brasileiras e seus estados, que tiveram mais prêmios pagos em dinheiro aos jogadores sortudos.

Entrada

A entrada é composta por um único caso de teste com 436 linhas, cada linha trazendo informações de um concurso e de qual cidade/estado é(são) o(s) ganhador(es). Os dados disponíveis em cada linha são: número do concurso (N), a data do sorteio (D), o número de ganhadores da Sena naquela cidade (G), a cidade, o estado e o valor do rateio (V) pago a cada ganhador. Onde: $1 \leq N \leq 2135$, $27/05/2009 \leq D \leq 30/03/2019$, $1 \leq G \leq 5$ e $R\$ 729.819,96 \leq V \leq R\$ 205.329.753,89$. Todos os dados separados por um ponto-e-vírgula. As cidades são pertencentes aos estados brasileiros, podendo existir cidades com o mesmo nome em estados diferentes. As entradas deverão ser lidas da entrada padrão. As entradas encerram-se com uma linha contendo o número 0.

Saída

Como saída, seu programa terá que mostrar 3 linhas, contendo o nome da cidade, o estado e o total de prêmios pagos (P) aos seus moradores, em cada linha, separados por um espaço. Onde $100000000 \leq P \leq 1000000000$ (valores inteiros, desprezar as casas decimais). As linhas deverão estar em ordem do maior ao menor valor pago. As saídas deverão ser escritas na saída padrão.

Exemplo de Entrada

Exemplo de Saída

1000;27/05/2009;1;Cidade A;SP;1000000.0	Cidade A SP 4452789
1002;15/04/2009;2;Cidade B;RJ;500000.5	Cidade C MG 4240700
1010;30/04/2009;2;Cidade C;MG;2120350.43	Cidade D AM 2431764
1020;31/05/2009;1;Cidade A;SP;3452789.67	
1040;30/06/2009;1;Cidade D;AM;2431764.1	
0	

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema C Consumo de Combustível

Arquivo fonte: Consumo.{ c | cc | java | py2 | py3 }
Autores: Prof. Hamilton Machiti (ETEC São Bernardo do Campo) e
Prof. Henrique Louro (ETEC de Caraguatatuba)

Tarefa

Em um contexto econômico pouco favorável, para ter sucesso nos negócios e driblar os efeitos da crise, as empresas precisam encontrar meios de garantir um maior controle das operações e reduzir custos ao máximo. Para os frotistas, diminuir gastos, significa diminuir o consumo de combustível, considerando que este consumo fica atrás apenas dos desgastes naturais. O combustível é a segunda maior despesa com as frotas de veículos das empresas. E, como estamos vivendo em plena crise do petróleo, melhorar o controle de combustível da sua frota é essencial para diminuir os custos e obter mais lucro para a sua empresa.

Então, estima-se que a utilização de um software que controle os gastos com combustível auxilie na redução do consumo entre 15% e 20%.

Sua tarefa é criar um programa, que dado os Kms rodados e a quantidade de combustível gasto, por veículo, em um determinado período, calcule o consumo médio individual, e o de toda frota. Por fim, liste os veículos em que o consumo no período esteja acima da média.

Entrada

A entrada é composta de um único caso de teste. A primeira linha contém um inteiro V que representa a quantidade de veículos da frota ($1 \leq V \leq 100$). As linhas seguintes contêm as informações de cada veículo no período. Os veículos não se repetem. Uma String representando a Placa, mais dois valores reais, um Km ($1 \leq Km \leq 2000$) representando a quantidade de quilômetros rodados, e um L ($1 \leq L \leq 800$) representando a quantidade de litros gastos. Todos os dados separados por um espaço. A entrada deverá ser lida da entrada padrão e encerra-se com uma linha contendo o número 0.

Saída

Como saída, seu programa terá que mostrar N linhas, contendo as placas dos veículos que estão acima da média de consumo da frota, na ordem que aparecem na entrada. As saídas deverão ser escritas na saída padrão.

Exemplo de Entrada

6 FAE6871 600.00 50.00 EWE2202 450.00 45.00 KAV1005 375.00 40.00 LWR2204 535.00 35.00 MMN3345 650.00 55.00 NEI9875 465.00 50.00 0
--

Exemplo de Saída

EWE2202 KAV1005 NEI9875

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema D Receita de Bolo

Arquivo fonte: Receita.{ c | cc | java | py }
Autores: Prof. Hamilton Machiti (ETEC São Bernardo do Campo) e
Prof. Henrique Louro (ETEC de Caraguatatuba)

Tarefa

Receita é um termo que tem vários conceitos e significados diversos dentro do que se pode definir ou conceituar. É uma palavra amplamente difundida na língua portuguesa e é de fácil explicação no âmbito global dos seus significados. Um dos significados ou conceitos do termo receita é a fórmula escrita e detalhada para se fazer um alimento, ou seja, uma fórmula de fabricação com todos os detalhes descritos na mesma para que se possa obter um prato ou uma bebida específica, seguindo uma série de instruções que resultam finalmente no que se vai degustar, conforme Figura 2 abaixo.

Joaquim é dono de uma padaria especializada em fabricar bolos. Mas ele está com um grande problema: não consegue controlar quanto gasta de ingredientes para fazer todos os bolos que vende em um período. Sabendo que você está estudando programação de computadores, pediu que fizesse um programa que o ajude-se nessa tarefa. Dadas as duas receitas dos bolos que fabrica e a quantidade fabricada diariamente de cada um, seu programa deverá informar a quantidade geral de ingredientes utilizados para fabricá-los.

RECEITAS		
Ingredientes	Bolo de Milho	Bolo de Coco
	Quantidade	Quantidade
Milho verde	200g	-
Óleo vegetal	200ml	120ml
Açúcar	250g	360g
Fubá	200g	-
Ovos	4	4
Farinha de trigo	15g	240g
Coco ralado	15g	100g
Fermento em pó	5g	10g
Leite de coco	-	200ml

Entrada

A entrada é composta de um único caso de teste. A primeira linha contém um inteiro D que representa a quantidade de dias de fabricação ($1 \leq D \leq 180$). As D linhas seguintes trazem dois números inteiros que representam as quantidades de bolos de milho (BM) e de coco (BC) fabricados por dia ($1 \leq BM, BC \leq 1000$). Todos os dados separados por um espaço. As entradas deverão ser lidas da entrada padrão. As entradas encerram-se com uma linha contendo o número 0.

Saída

Como saída, seu programa terá que mostrar 9 linhas, contendo a descrição (com acentos e cedilha suprimidos para compatibilidade dos sistemas) e a quantidade necessária de cada um dos ingredientes para realizar fabricação dos bolos, na ordem que aparecem na receita e separados por um espaço. As quantidades deverão ser seguidas das suas unidades de medidas, sem espaço, convertidas em Kg, para peso e L, para volume, quando houver. As saídas deverão ser escritas na saída padrão.

**CADERNO DE PROBLEMAS
MARATONA DE PROGRAMAÇÃO 2019-1**

Exemplo de Entrada

5
100 200
150 300
500 250
200 400
250 250
0

Exemplo de Saída

Milho verde 240Kg
Oleo vegetal 408L
Acucar 804Kg
Fuba 240Kg
Ovos 10400
Farinha de trigo 354Kg
Coco ralado 158Kg
Fermento em po 20Kg
Leite de coco 280L

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema E Joquempô

Arquivo fonte: Joquempo.{ c | cc | java | py2 | py3 }
Autores: Prof. Hamilton Machiti (ETEC São Bernardo do Campo) e
Prof. Henrique Louro (ETEC de Caraguatatuba)

Tarefa

Dois jogadores ficam frente a frente, com uma das mãos nas costas. Eles falam "joquempô" bem devagar e mostram a mão que está escondida, com o formato do objeto que escolheram, que pode ser: tesoura (dedos indicador e médio em forma de "V"); papel (mão aberta) e pedra (mão fechada). Ganha quem escolher o objeto que elimina o outro, seguindo as regras do vencedor: "a pedra quebra a tesoura", "a tesoura corta o papel" e "o papel embrulha a pedra".

Como este jogo é viciante, João e Pedro decidiram realizar vários torneios entre eles, para saber quem é o maior ganhador. Assim, precisam criar uma rotina que os ajude a calcular esses torneios, relatando ao final, quem foi o ganhador e qual foi a sua pontuação.

Sabendo que você é um programador avançado, pediu ajuda nessa empreitada. Sua tarefa será desenvolver um algoritmo que receba várias jogadas. Você deverá encontrar em cada jogada quem foi o vencedor e atribuir uma pontuação, para que no final seja dado o vencedor. Depois, deverá mostrar a soma dessas pontuações em um único número inteiro e o nome do ganhador.

Entrada

A entrada é formada por vários casos de testes. A primeira linha contém um inteiro J que representa a quantidade de jogadas ($1 \leq J \leq 100$). As J linhas seguintes trarão as informações de cada jogada, contendo: o nome de um jogador (N), sua jogada (E), o nome do outro jogador (N) e sua jogada (E), separados por um espaço, onde: $N = \{\text{João, Pedro}\}$ e $E = \{\text{pedra, tesoura, papel}\}$. As entradas deverão ser lidas da entrada padrão. As entradas encerram-se com uma linha contendo um número 0.

Saída

Como saída, seu programa terá que mostrar 1 linha, contendo qual é o jogador vencedor e quantas jogadas ele ganhou, separados por um espaço. Caso ocorra um empate a palavra "Empate" deverá ser mostrada. As saídas deverão ser escritas na saída padrão.

Exemplo de Entrada

```
5
Joao tesoura Pedro pedra
Pedro papel Joao tesoura
Pedro tesoura João papel
Joao pedra Pedro papel
Joao papel Pedro tesoura
3
Joao pedra Pedro papel
Pedro tesoura Joao pedra
Pedro papel Joao tesoura
2
Joao pedra Pedro papel
Pedro papel Joao tesoura
0
```

Exemplo de Saída

```
Pedro 4
Joao 2
Empate
```

(*) **Observação:** Os acentos deverão ser suprimidos para compatibilidade dos sistemas.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Problema F Bolinha de Sabão

Arquivo fonte: Bolinha.{ c | cc | java | py2 | py3 }
Autores: Prof. Tiago Jesus de Souza (ETEC Campinas)

Tarefa

Ana é uma menina que adora brincar de bolinhas de sabão em frente à sua casa. Todos os dias após a escola, ela fica lá soltando bolinhas junto com seu pai, uma pessoa muito curiosa e estudiosa. O pai olhando as quantidades de bolinhas que a sua filha soltava por vez, percebeu que esses valores poderiam representar o fatorial de um dos primeiros 8 números primos, aleatoriamente.

Na matemática utiliza-se equações para resolver uma série de situações-problema. Existem fórmulas para resolver alguns tipos de equações; padrões e procedimentos para resolução de outras; o uso de propriedades operatórias para simplificação de cálculos; enfim, utiliza-se todo conhecimento matemático a fim de obter o valor da incógnita em uma equação.

Em determinadas equações aparecem o conceito de fatorial, mostrando a necessidade de se conhecer bem a definição de fatorial e suas propriedades operatórias.

O conceito de fatorial é muito utilizado no estudo de arranjos e permutações, a fim de facilitar os cálculos. A ideia é bastante simples e de fácil compreensão.

O **fatorial** de um número inteiro **m** não negativo, é indicado por **m!** (lê-se “m fatorial”) e é definido pela relação:

$$m! = m \cdot (m-1) \cdot (m-2) \cdot (m-3) \dots 3 \cdot 2 \cdot 1, \text{ para } m \geq 2.$$

Algumas definições são:

$$1! = 1$$

$$0! = 1$$

Exemplos:

$$3! = 3 \cdot 2 \cdot 1 = 6$$

$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

Números primos são os números naturais que têm apenas dois divisores: o 1 e ele mesmo.

Exemplo 1:

- 1) 1 não é primo, pois só é divisível por ele mesmo;
- 2) 2 tem apenas os divisores 1 e 2, portanto 2 é um número primo;
- 3) 17 tem apenas os divisores 1 e 17, portanto 17 é um número primo.
- 4) 10 tem os divisores 1, 2, 5 e 10, portanto 10 não é um número primo.

Exemplo 2: O número 113:

- 1) Não é par, portanto, não é divisível por 2;
- 2) $1+1+3 = 5$, portanto, não é divisível por 3;
- 3) Não termina em 0 nem em 5, portanto não é divisível por 5;
- 4) Por 7: $113 / 7 = 16$, com resto 1. O quociente (16) ainda é maior que o divisor (7).
- 5) Por 11: $113 / 11 = 10$, com resto 3. O quociente (10) é menor que o divisor (11). Além disso o resto é diferente de zero (o resto vale 3), portanto 113 é um número primo.

Sua tarefa é criar um programa, que dado a quantidade de bolinhas que a menina Ana soltou, verificar se esse número corresponde ao fatorial de um dos 8 primeiros números primos.

CADERNO DE PROBLEMAS MARATONA DE PROGRAMAÇÃO 2019-1

Entrada

A entrada é composta por vários casos de teste, cada linha trazendo a informação referente a quantidade de bolinhas de sabão que a menininha soltou. As entradas deverão ser lidas da entrada padrão. As entradas encerram-se com uma linha contendo o número 0.

Saída

Como saída, seu programa terá que mostrar, para cada entrada válida, uma linha contendo o cálculo do fatorial do número de entrada, conforme exemplo abaixo. Deverá haver um espaço entre o operador de multiplicação e os números, bem como entre eles e o sinal de igualdade. As saídas deverão ser escritas na saída padrão.

Exemplo de Entrada Exemplo de Saída

2	$2! = 2 \cdot 1 = 2$
4	$3! = 3 \cdot 2 \cdot 1 = 6$
6	$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$
120	$11! = 11 \cdot 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 39916800$
10	$17! = 17 \cdot 16 \cdot 15 \cdot 14 \cdot 13 \cdot 12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 355687428096000$
39916800	
12345678910	
355687428096000	
121532100408832000	
0	

BOA SORTE!

