



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Maratona de Programação

Seletiva UFPE 2016

1ª Sessão de Warm-up
Gabarito das questões

Maratona de Programação CIn-UFPE
22 de Fevereiro de 2016

A - Hashmat the Brave Warrior

Hashmat é um guerreiro cujo seu grupo de soldados se move de um local a outro para lutar contra os seus oponentes. Antes da luta, ele apenas calcula uma coisa: diferença entre a quantidade de soldados que possui e a quantidade de soldados oponentes. A partir desta diferença ele decide se vai ou não lutar. Às vezes, Hashmat tem mais soldados do que o seu oponente, mas, na maioria das vezes, não.

Entrada:

A entrada contém dois números inteiros em cada linha. Estes dois números denotam respectivamente a quantidade de soldados do exército de Hashmat e do seu oponente. Nenhum número de entrada é maior do que 2^{32} . A entrada termina com fim de arquivo (EOF).

Saída:

Para cada linha de entrada imprima a diferença entre o número de soldados de Hashmat e do seu oponente. Cada saída deve ser impressa em uma linha separada.

Exemplo:

Entrada:	Saída:
10 12	2
10 14	4
100 200	100

A - Hashmat the Brave Warrior - Solução

Ideia:

A ideia da questão é ler os dois valores e imprimir a diferença absoluta entre eles.

O cuidado é com o fato do número poder ser até 2^{32} .

Recomendo *long long* para poder fazer a subtração. Pode também ser feita com *unsigned int* tratando com cuidado para a subtração não dar negativo.

Pode também ser utilizado a função `abs()` para calcular a diferença absoluta.

Solução:

```
#include <bits/stdc++.h>

using namespace std;

long long A,B;

int main(){
    while(scanf("%lld%lld",&A,&B) == 2){
        long long ans = A-B;
        if(ans < 0) ans = -ans;
        printf("%lld\n", ans);
    }
}
```

B - Candy I

Jennifer é uma professora do primeiro ano do ensino fundamental. Ela saiu em uma excursão com a classe dela. Ela levou um pacote de balinhas para cada criança. Infelizmente, os tamanhos dos pacotes não são os mesmos.

Jennifer está com medo de que cada criança queira ter o maior pacote de balinhas e isso vai levar à disputas e até brigas entre as crianças. Ela quer evitar isso. No entanto, ela decidiu abrir todos os pacotes, contar as balinhas em cada pacote e mover algumas balinhas dos pacotes maiores para os pacotes menores, sendo assim, cada pacote vai conter a mesma quantidade de balinhas. A questão é quantas balinhas ela terá de mover.

Entrada:

A entrada consiste de vários blocos de dados. Cada bloco começa com um número de pacotes de balinhas N ($1 \leq N \leq 10.000$) seguido de N inteiros (cada um menor que 1.000) em linhas separadas, informando a quantidade de balinhas em cada pacote. Após o último bloco de dados tem um número -1.

Saída:

A saída deve conter uma linha com o menor número de movimentos para cada bloco de dados. Um movimento consiste em pegar uma balinha de um pacote e colocar em outro pacote. Se não é possível ter o mesmo número de balinhas em cada pacote, imprima o número -1.

Exemplo:

Entrada:	Saída:
5 1 1 1 1 6 2 3 4 -1	4 -1

B - Candy I - Solução

Ideia:

Para resolvermos a questão temos que entender o seguinte: para que seja possível dividir as balas em N saquinhos iguais é necessário que a soma das balas seja divisível por N.

Logo temos que ter a soma da quantidade de balas. Caso essa soma seja divisível por N, então podemos arranjar uma solução, caso contrário só imprimir -1.

Em caso de haver solução, para calcularmos a resposta é o seguinte, podemos contar o quanto cada saquinho que tem valor menor do que o valor esperado precisa e somamos cada diferença.

Solução:

```
#include <bits/stdc++.h>

using namespace std;

int N;
int ar[10010];

int main(){
    while(scanf("%d", &N) == 1 && N != -1){
        int total = 0;
        for(int n = 0; n < N; n++) {
            scanf("%d", &ar[n]);
            total += ar[n];
        }

        if(total%N == 0){
            int media = total/N;
            int ans = 0;
            for(int n = 0; n < N; n++){
                if(ar[n] < media){
                    ans += media-ar[n];
                }
            }
            printf("%d\n", ans);
        }else printf("-1\n");
    }
}
```

C - Cost Cutting

A companhia XYZ sofreu grandes perdas devido à recessão e está tomando várias medidas para reduzir seus gastos. Algumas das medidas incluem redução do espaço de trabalho, redução de incentivos, corte de benefícios e demissões de funcionários.

A companhia possui três (3) empregados trabalhando no departamento de contas e irá demitir dois (2) deles. Após uma série de reuniões, foi decidido que será demitido o empregado que recebe o maior salário, assim como o que recebe o menor. Isso é uma tendência comum durante épocas de crise como agora.

Serão dados a você os salários dos 3 empregados trabalhando no departamento de contas. Seu trabalho é determinar o salário da pessoa que continuará trabalhando na companhia.

Entrada:

A primeira linha da entrada é um inteiro T ($T < 20$) que indica o número de casos teste. Cada caso consiste de uma linha com 3 inteiros positivos distintos. Esses 3 inteiros representam os salários dos três empregados. Todos os inteiros estarão no intervalo [1000, 10000].

Saída:

Para cada caso teste, a saída será composta pelo número do caso teste, seguido pelo salário do empregado que não foi demitido.

Exemplo:

Entrada:	Saída:
3 1000 2000 3000 3000 2500 1500 1500 1200 1800	Case 1: 2000 Case 2: 2500 Case 3: 1500

C - Cost Cutting - Solução

Ideia:

Existem vários jeitos de resolver esta questão. O modo mais rápido é ordenando os três valores e pegando o valor do meio utilizando a função sort.

No entanto, existe um jeito simples de responder a questão que é somando todos os valores e subtraindo o valor máximo e o valor mínimo dos três, o que resta é o valor do meio.

Solução:

```
#include <bits/stdc++.h>

using namespace std;

int T;
int ar[10];

int maxVal(){
    int maxi = ar[0];
    for(int i = 0 ; i < 3 ; i++)
        maxi = max(maxi, ar[i]);
    return maxi;
}

int miniVal(){
    int mini = ar[0];
    for(int i = 0; i < 3; i++)
        mini = min(mini, ar[i]);
    return mini;
}

int main(){
    scanf("%d",&T);
    for(int t = 1; t <= T; t++){
        scanf("%d%d%d", &ar[0], &ar[1], &ar[2]);
        int mini = miniVal();
        int maxi = maxVal();
        int medium = ar[0]+ar[1]+ar[2] - mini - maxi;
        printf("Case %d: %d\n", t, medium);
    }
}
```

D - A Change in Thermal Unit

Medir temperaturas e diferenças de temperatura são tarefas comuns em muitas pesquisas e aplicações. Infelizmente, existe mais de uma unidade para medir temperaturas. Isso introduz muita confusão às vezes. Duas unidades de medida populares são Celsius (C) e Fahrenheit (F). A conversão de F para C é dada pela fórmula:

$$F = (9/5)*C + 32$$

Neste problema, você receberá uma temperatura inicial em C e um aumento de temperatura em F. Você terá que calcular a nova temperatura em C.

Entrada:

A entrada começa com um inteiro T ($T \leq 100$), indicando o número de casos de teste. Cada caso contém uma linha com dois inteiros C e d ($0 \leq C, d \leq 100$), onde C representa a temperatura inicial em Celsius e d representa o aumento de temperatura em Fahrenheit.

Saída:

Para cada caso, imprima o número do caso e a nova temperatura em Celsius depois de arredondar para a segunda casa decimal.

Exemplo:

Entrada:	Saída:
2 100 0 0 100	Case 1: 100.00 Case 2: 55.56

D - A Change in Thermal Unit - Solução

Ideia:

A questão é bem direta. Para resolver é necessário converter a temperatura inicial para Fahrenheit e então acrescentar o delta para então voltar a Celcius. Existem outras formas também.

Solução:

```
#include <bits/stdc++.h>

using namespace std;

int T, C, d;

int main(){
    scanf("%d", &T);
    for(int t = 1; t <= T; t++){
        scanf("%d%d", &C, &d);
        double inF = (9.0*C)/5.0 + 32.0 + d;
        double inC = ((inF-32.0)*5.0)/9.0;
        printf("Case %d: %.2lf\n", t, inC);
    }
}
```

E - Ecological Premium

Fazendeiros alemães são premiados dependendo das condições de sua propriedade rural. Imagine o seguinte regulamento simplificado: você sabe o tamanho, em metros quadrados, da propriedade rural de cada fazendeiro, assim como o número de animais criados em cada uma. Nós não iremos fazer distinção entre animais diferentes, apesar disso se afastar da realidade. Além disso, você é informado sobre o grau de uso na propriedade de práticas e equipamentos benéficos ao meio ambiente, sendo esse grau de preocupação ambiental expresso como um único inteiro maior que zero.

O valor da quantia em dinheiro que o fazendeiro irá receber pode ser calculado a partir desses parâmetros da seguinte forma: primeiramente, você precisará do espaço ocupado por cada animal, em média. Esse valor (em metros quadrados) será então multiplicado pelo parâmetro referente ao grau de preocupação ambiental do fazendeiro, falado anteriormente, e esse valor resultante é o prêmio dado ao fazendeiro por cada animal que possui.

Para computar a quantia final que o fazendeiro receberá, basta multiplicar o valor recebido por cada animal, pela quantidade de animais que o fazendeiro possui.

Entrada:

A primeira linha consiste em um único inteiro positivo n ($n < 20$), o número de casos teste. Cada caso teste começa com uma linha contendo um único inteiro f ($0 < f < 20$), o número de fazendeiros neste caso teste. Essa linha é seguida por uma linha para cada fazendeiro, contendo cada uma três inteiros positivos: o tamanho da propriedade rural em metros quadrados, o número de animais que o fazendeiro possui e o valor inteiro que expressa o grau de preocupação ambiental do fazendeiro. A entrada termina com o fim do arquivo. Nenhum inteiro na entrada é maior que 100000 ou menor que 0.

Saída:

Para cada caso teste imprima uma linha contendo um único inteiro que representa a quantia somada que o orçamento alemão deverá pagar aos fazendeiros, a qual sempre é um número inteiro. Não imprima nenhuma linha em branco.

Exemplo:

Entrada:	Saída:
3 5 1 1 1 2 2 2 3 3 3 2 3 4 8 9 2 3	38 86 7445

9 1 8 6 12 1 8 1 1 3 10 30 40 9 8 5 100 1000 70	
---	--

E - Ecological Premium - Solução

Ideia:

Esta é uma típica questão de enunciado grande, que pode confundir, e que você tem de capturar o que é essencial. Tudo que ele pede, na verdade, é que você multiplique a área total do fazendeiro pelo seu grau ambiental e soma cada um desses resultados dentro de um caso de teste.

Solução:

```
#include <bits/stdc++.h>

using namespace std;

int N, F, A, B, C;

int main(){
    scanf("%d",&N);
    while(N--){
        scanf("%d", &F);
        int total = 0;
        while(F--){
            scanf("%d%d%d",&A,&B,&C);
            total += A*C;
        }
        printf("%d\n", total);
    }
}
```