

Lista de Exercícios de Estruturas de Dados e Algoritmos

Q1) Implemente um programa que, infinitamente, receba, como parâmetro de entrada, um número n e retorne os n primeiros números primos existentes. Seu programa para quando n for menor ou igual a zero.

Q2) Implemente um programa que, infinitamente, receba, como parâmetro de entrada, um número n e retorne os n primeiros números primos existentes depois de n . Por exemplo, se $n = 2$, a resposta será os primos 3 e 5. É necessário salientar que n não precisa ser primo. Seu programa para quando n for menor ou igual a zero.

Q3) **[Problema 1103 do URI Online Judge]** Daniela é enfermeira em um grande hospital, e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores. Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar. No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso.

Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir.

Entrada: a entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros $H1$, $M1$, $H2$, $M2$, com $H1:M1$ representando a hora e minuto atuais, e $H2:M2$ representando a hora e minuto para os quais o alarme despertador foi programado ($0 \leq H1 \leq 23$, $0 \leq M1 \leq 59$, $0 \leq H2 \leq 23$, $0 \leq M2 \leq 59$). O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

Saída: para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Daniela tem para dormir.

Entrada	Saída
1 5 3 5	120
23 59 0 34	35
21 33 21 10	1417
0 0 0 0	

Q4) **[Problema 1028 do URI Online Judge]** Ricardo e Vicente são aficionados por figurinhas. Nas horas vagas, eles arrumam um jeito de jogar um “bafo” ou algum outro jogo que envolva tais figurinhas. Ambos também têm o hábito de trocarem as figuras repetidas com seus amigos e certo dia pensaram em uma brincadeira diferente. Chamaram todos os amigos e propuseram o seguinte: com as figurinhas em mãos, cada um tentava fazer uma troca com o amigo que estava mais perto seguindo a seguinte regra: cada um contava quantas figurinhas tinha. Em seguida, eles tinham que dividir as figurinhas de cada um em pilhas do mesmo tamanho, no maior tamanho que fosse possível para ambos. Então, cada um escolhia uma das pilhas de figurinhas do amigo para receber. Por exemplo, se Ricardo e Vicente fossem trocar as figurinhas e tivessem respectivamente 8 e 12 figuras, ambos dividiam todas as suas figuras em pilhas de 4 figuras (Ricardo teria 2 pilhas e Vicente teria 3 pilhas) e ambos escolhiam uma pilha do amigo para receber.

Entrada: a primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 3000$), indicando o número de casos de teste. Cada caso de teste contém 2 inteiros $F1$ ($1 \leq F1 \leq 1000$) e $F2$ ($1 \leq F2 \leq 1000$) indicando, respectivamente, a quantidade de figurinhas que Ricardo e Vicente têm para trocar.

Saída: para cada caso de teste de entrada haverá um valor na saída, representando o tamanho máximo da pilha de figurinhas que poderia ser trocada entre dois jogadores.

Entrada	Saída
3 8 12 9 27 259 111	4 9 37

Q5) Supondo as seguintes declarações:

```
int x, y, z;
```

```
int teste;
```

```
x = 5;
```

```
y = x++;
```

```
z = x--;
```

Informe os valores de **teste**, **x**, **y** e **z** depois da avaliação das seguintes expressões:

(a) `teste = !y == !x;`

(b) `teste = ((x++ > y) || (--z <= y)) & z ^ x;`

(c) `teste = ((!x) || (!(!z)) && ++x);`

(d) `teste = ((x + y) > z) & (x++);`

(e) `teste = x & y + !3 | 4 | --z;`

(f) `y >>= 1;`

(g) `x <<= 2;`