

Esta é uma lista complementar de exercícios. A resolução é individual. A pontuação obtida com a resolução desta lista acrescenta, no máximo , 2,0 pontos ao resultado dos dois exercícios escolares.

- (0,8) 1. Você foi contratado para implementar um sistema de administração aeroportuário que facilite o trabalho dos controladores de voo. Para tal fim, o seu sistema deve ser capaz de automaticamente avisar aos aviões quando eles podem decolar ou aterrissar. O programa irá receber como entrada uma quantidade N (especificada pelo usuário) de aviões esperando para sair. Depois disso, ele irá receber, para cada avião, a hora de saída do mesmo (para fins de teste, essa hora de saída será especificada em milissegundos após o início do programa). Após ter recebido os aviões que irão decolar, o programa irá receber um número M (também especificado pelo usuário) de aviões que irão chegar, cada um com a sua hora esperada de chegada. Finalmente, o usuário deverá inserir um número K , correspondente ao número de pistas disponíveis no aeroporto.

Quando chega a hora de um avião decolar ou aterrissar, o programa deve verificar se há uma pista livre. Se houver, o avião irá utilizar a pista. O processo de decolagem ou de aterrissagem de um avião ocupa a pista por 500 milissegundos. Caso não haja uma pista livre, o avião deverá aguardar até que haja.

Quando um avião consegue decolar ou aterrissar com sucesso, o programa deve imprimir o horário esperado de saída do avião, o horário real, e o atraso que ocorreu. Além disso, para evitar prejuízos para os aeroportos, seu programa deve minimizar os atrasos - portanto, quando uma pista for ocupada, deve ser dada prioridade ao avião cujo horário de saída é mais cedo.

Por motivos de eficiência, o programa deve ser implementado sem uso de espera ocupada, e quando um avião for ser notificado de que há uma pista livre, deve ser realizada uma única chamada.

- (0,6) 2. Crie N jogadores e um vetor com $N-1$ Cadeiras, cada cadeira é um tipo atômico. A cada rodada, todos os jogadores tentarão (aleatoriamente) sentar em alguma cadeira vazia. Eles continuam tentando até todas as cadeiras estarem ocupadas. Em seguida, é mostrado na tela o único jogador que não conseguiu sentar "O jogador X foi eliminado", onde X é o seu nome. Todas as cadeiras ficam vagas novamente e uma é retirada do jogo. Inicia-se uma nova rodada. O programa acaba quando só sobrar 1 jogador, e deverá ser mostrado na tela "O jogador X foi o vencedor! "

- (0,3) 3. Escreva uma função `fibs` que retorna um *closure iterator* que produz os primeiros n números da sequência de Fibonacci. Por exemplo,

```
s = 0
for i in fibs(5) do
    s = s + i
end
print(s) -- 12
```

- (0,3) 4. Número triangulares (https://en.wikipedia.org/wiki/Triangular_number) constituem uma sequência de inteiros em que o primeiro número é 1. O n -ésimo número

triangular é encontrado tomando o número anterior e adicionando n. Escreva uma função (closure) para números triangulares. Exemplo

```
for x in num_triangu(6) do
    io.write(x, " ")
end
```

— imprime: 1 3 6 10 15 21