



1. Considere n processos a serem executados (numerados de 1 a n), uma fila F contendo a *ordem de chamada* dos processos (i.e., o processo no início da fila foi o primeiro chamado e aquele no final foi o último) e uma pilha P contendo a *ordem de execução* ideal dos processos (i.e., se um processo i está sobre outro processo j em P , então i deve ser executado antes de j). Supondo que cada mudança de ordem dos processos na fila, bem como a execução de um processo, consome 1 segundo, crie a função $ttm(F, P)$, que recebe a fila F e a pilha P e devolve o tempo total mínimo necessário para executar todos os processos na ordem ideal. Por exemplo, se a fila F contém os processos 2, 3 e 1 (sendo 2 o processo no início de F e 1 aquele no final de F) e a ordem de execução ideal em P é 1, 3 e 2 (sendo 1 o processo no topo de P e 2 aquele na base de P), então o tempo total mínimo necessário para executar todos os processos na ordem ideal é 6 segundos. Note que a função $ttm()$ deve funcionar para qualquer quantidade de processos e, no final da execução da função, F e P devem estar vazias.

```
#include <stdio.h>
#include "pilha.h"
#include "fila.h"

int ttm(Fila F, Pilha P) { // tempo total mínimo
    ...
}

int main(void) {
    Fila F = fila(3);
    enfileira(2,F);
    enfileira(3,F);
    enfileira(1,F);

    Pilha P = pilha(3);
    empilha(2,P);
    empilha(3,P);
    empilha(1,P);

    printf("\nTempo total mínimo: %d\n\n", ttm(F,P) );

    destroif(&F);
    destroip(&P);

    return 0;
}
```

2. Uma cadeia c composta apenas por letras A e B é uma *concha* se cada letra em c pode ser conectada a exatamente uma *única* outra letra igual a ela em c , por meio de um arco, de modo que nenhum arco intercepte outro arco. Por exemplo, na figura a seguir, a cadeia em (a) é uma concha, mas aquela em (b) não o é.

"AAABAAABBABAAA"

(a) Essa cadeia é uma concha

"BBBAAAAABA"

(b) Essa cadeia **não** é uma concha

Com base nesta definição:

- Crie a função `concha(c)`, que recebe uma cadeia c e devolve 1 se c for uma concha (caso contrário, devolve 0).
- Crie a função `filtra(F)`, que recebe uma fila F contendo cadeias compostas apenas por letras A e B e exhibe apenas aquelas cadeias em F que são conchas, bem como o *total* de cadeias em F que são conchas.



```
#include <stdio.h>
#include <string.h>
#include "pilha.h"
#include "fila.h"

int concha(char *c) { // verifica se c é uma concha
    ...
}

void filtra(Fila F) { // exibe apenas as cadeias em F que são conchas
    ...
}

int main(void) {
    Fila F = fila(20);
    enfileira("AAAAAAAAABAAAABA", F);
    enfileira("BBBBBBBBBBBBBBB", F);
    enfileira("ABBABBABBABB", F);
    enfileira("AAABAAABBABAAA", F);
    enfileira("ABAABAAAAAAAAA", F);
    enfileira("AABAAAAAABAAAA", F);
    enfileira("AAAABB", F);
    enfileira("AABBBBAABAAB", F);
    enfileira("ABAAAAAAAAAAB", F);
    enfileira("ABABBBBABAB", F);
    enfileira("BBBBBBBBBBBBBBB", F);
    enfileira("AABAABAAAA", F);
    enfileira("AABAAABBBBBABB", F);
    enfileira("BBBAAAAABA", F);
    enfileira("BBBBBBBBBBBBBBB", F);
    enfileira("AAAAAAAAAAAAABB", F);
    enfileira("BAAAAABAAAAAAA", F);
    enfileira("BBBBABBBBBBA", F);
    enfileira("BBABABAAAAAABB", F);
    enfileira("ABAABAABAB", F);

    filtra(F); // deve exibir 12 conchas

    destroif(&F);
    return 0;
}
```