

## EXERCÍCIO PRÁTICO 2

Enunciado baseado no trabalho proposto na disciplina "Algoritmos e Estruturas de Dados" da Universidade Federal de Ouro Preto (<http://www.decom.ufop.br/menotti/aedI092/tps/tp2.pdf>).

### INSTRUÇÕES:

- A atividade deve ser realizada em duplas e entregue até às 23h59 do dia **20/02/2013** via Moodle, por apenas um integrante da dupla.
- A atividade deve ser entregue na forma de um arquivo compactado nomeado da seguinte forma nome1\_nome2.zip ou nome1\_nome2.rar, onde nome1 e nome2 são os nomes dos integrantes da dupla.
- Faça o exercício apenas com a sua dupla.
- Endentem o código, deem nomes de variáveis que reflitam suas funções e façam comentários pertinentes no código.
- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais estará.
- Em caso de dúvida entre em contato por email [alinemello@unipampa.edu.br](mailto:alinemello@unipampa.edu.br) ou pessoalmente na sala 229.

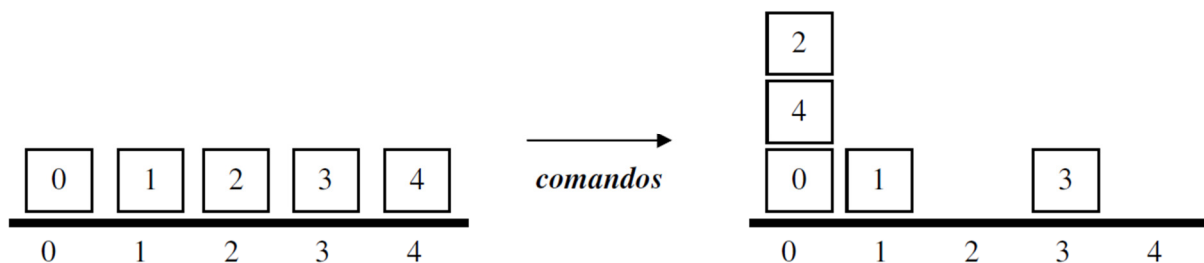
### AValiação:

O EP2 vale 10 pontos e tem peso igual ao de uma prova, ou seja 1,67 na nota final da disciplina. A nota máxima do trabalho perde 1 ponto a cada dia de atraso na entrega. Por exemplo, caso o trabalho seja entregue no dia 21/02/2013, a nota máxima será 9.

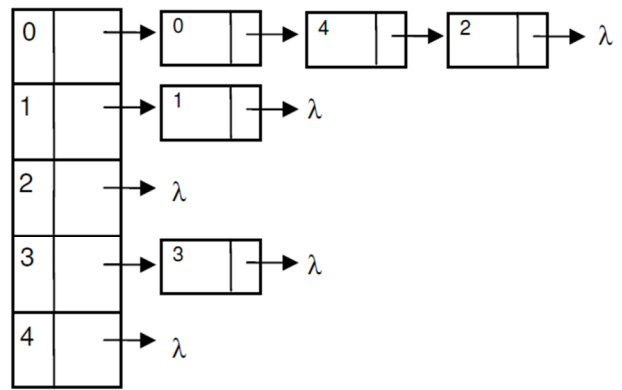
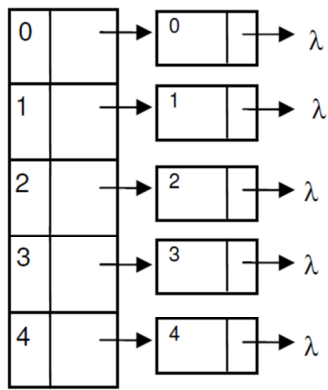
Trabalhos desenvolvidos com **listas duplamente encadeadas receberão 1 ponto extra**.

### ENUNCIADO:

Várias áreas da ciência da computação usam domínios simplificados para abstrair diversos tipos de problemas. Por exemplo, algumas das primeiras pesquisas de inteligência artificial nas áreas de planejamento e robótica eram feitas utilizando o "mundo dos blocos", no qual um braço robótico realizava tarefas simuladas envolvendo a manipulação de blocos [1]. Nesse trabalho você vai implementar um mundo de blocos bem simples [2], que vai funcionar de acordo com certas regras e obedecer comandos de movimentação de blocos dados pelo usuário, simulando o que seria a manipulação através um braço robótico. O seu mundo de blocos começa com cada bloco na sua posição inicial, e depois de uma série de comandos deve terminar em uma configuração final, como mostrado da figura abaixo para 5 blocos (n=5).



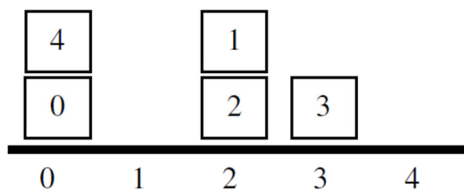
Para implementar o seu mundo de blocos, você vai criar um tipo abstrato de dados chamado Tblocos que é composto por um vetor de listas encadeadas (onde cada lista corresponde a um monte de blocos e cada célula vai corresponder a um bloco) e pelas funções para a manipulação dessa estrutura. Por exemplo, as duas configurações mostradas acima estariam implementadas da seguinte forma utilizando o seu TAD:



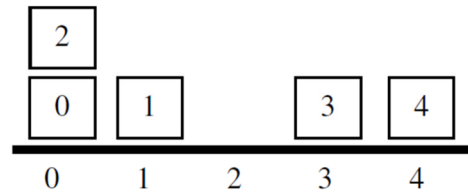
Onde  $\lambda$  corresponde à NULL.

Os comandos possíveis para a manipulação do seu mundo de blocos são:

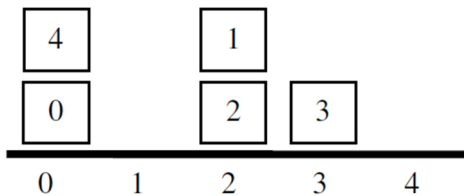
- **move a onto b:** Move o bloco a para cima do bloco b retornando eventuais blocos que já estiverem sobre a ou b para as suas posições originais.



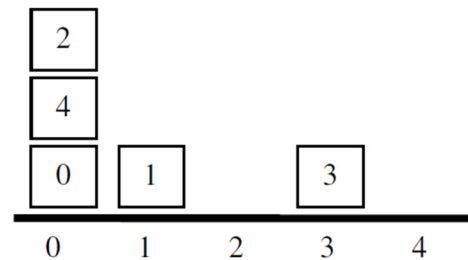
*move 2 onto 0*



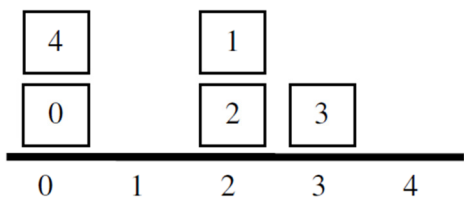
- **move a over b:** Coloca o bloco a no topo do monte onde está o bloco b retornando eventuais blocos que já estiverem sobre a às suas posições originais.



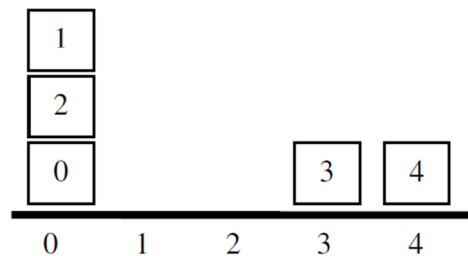
*move 2 over 0*



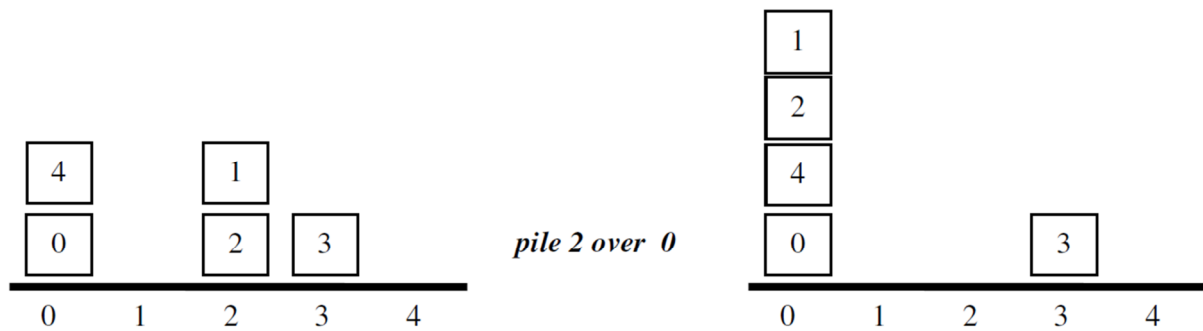
- **pile a onto b:** Coloca o bloco a juntamente com todos os blocos que estiverem sobre ele em cima do bloco b, retornando eventuais blocos que já estiverem sobre b as suas posições originais.



*pile 2 onto 0*



- **pile a over b:** Coloca o bloco a juntamente com todos os blocos que estiverem sobre ele sobre o monte que contem o bloco b.



- **quit: termina a execução.**

Comandos onde  $a = b$  ou onde  $a$  e  $b$  estejam no mesmo monte devem ser ignorados.

Além de funções para a implementação desses comandos, crie funções auxiliares no seu TAD para simplificar a implementação e evitar a repetição de código. Por exemplo, uma função para retornar um ou mais blocos para as suas posições originais vai ser utilizada por quase todos os comandos.

O seu programa principal deverá ser basicamente um parser que lê comandos a partir de um arquivo de entrada, executa os comandos no mundo dos blocos e imprime a saída (configuração final do mundo de blocos) para um arquivo. O programa pode ainda imprimir a representação do mundo dos blocos após cada comando. Os arquivos de entrada e saída devem obedecer os seguintes formatos:

- **Arquivo de entrada:** é composto pelo número  $n$  de blocos na primeira linha, uma sequência de comandos (um por linha) e o comando quit na última linha. Você pode considerar que não haverá erros de sintaxe no arquivo, ou seja, os comandos vão estar escritos corretamente.
- **Arquivo de saída:** o arquivo de saída vai representar a configuração final do seu mundo de blocos. Em cada linha deve estar o número da posição original, seguida de um dois pontos (:) e a lista de blocos que estão naquela posição com um espaço entre eles. Esse formato deve ser rigorosamente seguido.

Para o primeiro exemplo mostrado, os arquivos de entrada e saída são listados abaixo. Note que vários arquivos de entrada podem levar a mesma configuração final.

#### Entrada:

```
5
move 3 onto 0
move 2 over 4
pile 4 onto 0
quit
```

#### Saída:

```
0: 0 4 2
1: 1
2:
3: 3
4:
```

Para executar seu programa, use parâmetros de linha de comando para fazer sua chamada. Por exemplo, se o seu programa chama-se EP2 e você quiser executar o programa com a entrada arq1.txt gerando a saída para o arquivo saida.txt

**EP2 arq1.txt saida.txt**

#### **REFERÊNCIAS:**

[1] Artificial Intelligence – A Modern Approach. Stuart Russell and Peter Norvig, 2nd Edition, Prentice Hall, 2003.

[2] The Blocks Problem – <http://online-judge.uva.es/p/v1/101.html>.