

MC102 - Algoritmos e Programação de Computadores**Turmas QRSTWY****Instituto de Computação - Unicamp****Professores:** Hélio Pedrini e Zanoni Dias**Monitores:** Andre Rodrigues Oliveira, Gustavo Rodrigues Galvão, Javier Alvaro Vargas Muñoz e Thierry Pinheiro Moreira

Lab 07a - Cargo-Bot - Part II

Prazo de entrega: 04/05/2015 às 13h59m59s**Peso:** 5

Após a aprovação da primeira versão do jogo [Cargo-Bot](#) feita por você, os responsáveis pela [TwoLivesLeft](#) lhe desafiaram a adicionar mais funcionalidades do jogo original. Os comandos básicos, a detecção de um acidente e o funcionamento do braço mecânico continuam como descritos no [Laboratório 04a](#).

Agora, ao invés de apenas um programa, ou seja, apenas uma sequência de comandos, a nova versão do Cargo-Bot poderá conter até 9 programas diferentes, que serão executados numa sequência pré-determinada. Cada programa conterá apenas comandos básicos ou condicionais (considere que um programa nunca chama outro).

Além disso, as caixas agora podem ser de três cores diferentes: azul, que será representada pela letra B (blue), verde, que será representada pela letra G (green) e amarelo, que será representada pela letra Y (yellow). Por exemplo, a configuração de caixas da imagem abaixo, supondo que o número máximo de caixas que podem ser empilhadas é 6, será representada da seguinte forma em sua versão do Cargo-Bot:



```
.....  
.B...B.  
.B.G.G.  
.GBB.G.  
.BGBGG.  
.GBGBB.
```

Note que a ausência de caixas em uma posição é representada pelo caractere ponto ".". Além disso, note que o número de linhas da representação é igual ao número máximo de caixas que podem ser empilhadas.

Por fim, a nova versão do Cargo-Bot poderá utilizar condicionais para realização dos comandos básicos dos programas, conforme explicado anteriormente. Os comandos condicionais podem ser:

- B,G,Y: realiza o comando caso a caixa que o braço mecânico segura seja da mesma cor que o condicional;
- A: realiza o comando caso o braço mecânico esteja segurando uma caixa qualquer;
- N: realiza o comando básico caso o braço mecânico não esteja segurando uma caixa.

Os comandos básicos e os comandos condicionais de um programa serão representado numa única linha, sendo que o comando básico seguido ao condicional só deve ser executado se a condição testada for verdadeira. Por exemplo, os comandos do programa da imagem abaixo serão representados da seguinte forma em sua versão do Cargo-Bot:



DLBDRRYDL*

Entrada

- A primeira linha da entrada consistirá de 5 números inteiros, $X Y P C E$, onde:
 - X representa o número de posições existentes para empilhamento das caixas, com $2 \leq X \leq 10$;
 - Y representa o número máximo de caixas que podem ser empilhadas, com $2 \leq Y \leq 10$;
 - P representa o número de programas existentes, com $1 \leq P \leq 9$;
 - C representa o número máximo de comandos (básicos e condicionais) em um programa, com $1 \leq C \leq 60$;
 - E representa o número de execuções de programas, com $1 \leq E \leq 60$.
- As próximas Y linhas conterão X caracteres cada, representando o estado inicial das caixas;
- A próxima linha, contendo E dígitos entre 1 e P , representará a ordem em que os programas serão executados. Note que um programa pode ser executado múltiplas vezes.
- Por fim, existirão P linhas, uma para cada programa, com no máximo C caracteres representando os comandos (básicos e condicionais) e um caractere extra, *, indicando fim da lista de comandos.
- Assim como na primeira versão do Cargo-Bot, considere que, antes do primeiro comando do primeiro programa, o braço robótico encontra-se na posição mais à esquerda e não segura nenhuma caixa.

Saída

- Se um acidente ocorreu durante a execução dos comandos, seu programa deve imprimir apenas "Um acidente ocorreu".
- Caso contrário, seu programa deve imprimir a configuração final das caixas após a execução de todos os comandos, com Y linhas e X colunas.

Exemplos

#	Entrada	Saída
1	7 6 7 20 38B...B. .B.G.G. .GBB.G. .BGBGG. .GBGBB. 21212721276167676761365232364246464256 B...BG BG...BG BG...BG BG...BG BG...BG BG...BG

	BLBLBLBLBLBLBD* LLLLLLRD* LLLLLLRRND* LLLLLLRRRND* RRRRRRLND* RRRRRRLD* GRGRGRGRGRGD*	
2	3 3 1 10 10 Y.. G.. B.. 1111111111 DARBRNL*	..Y ..G ..B
3	7 6 4 12 25 BBB.... BBB.... BBB.... BBB.... 1121212312121212412121212 DNR* BRBRBRDLLL* LLLLLR* LLLLLRR*	Um acidente ocorreu
4	6 7 2 10 19 G..... G..... G..... G..... G..... G..... 1111121111211121121 DRGDNL* NR* GGGGGG
5	7 6 3 21 25Y.... ..Y...Y ..YY...Y 2221222122212222132213221 D* NRNDALALAL* NRNDNRNDNRNDNRNDNRND*	Y..... Y..... Y..... Y..... Y..... Y.....