Problema: (BOCA: L6_4e5_2c) Imagine o mesmo cenário das questões *a* e *b*. Porém agora, você já leu as cartelas da partida conforme a questão anterior. A sua terceira tarefa é simular a execução de várias partidas na sequência utilizando um mesmo grupo de cartelas. Para isso, serão dadas as *n* cartelas das partidas e *p* sequências de números a serem "sorteados" em cada partida. O programa deverá simular o sorteio desses números e anunciar o resultado da partida assim que houver algum ganhador para alguma pedra. As pedras restantes devem ser lidas para limpar a sequência corrente, mas não devem ser contabilizadas nas cartelas ou na partida posterior. Quando uma sequência finalizar, então a próxima partida começa utilizando as mesmas cartelas.

- Entrada: Um grupo de cartelas, idêntico ao da questão *b*, seguido do número de sequências *p* (inteiro) e, posteriormente, dos números das sequências em si. Cada sequência terá um conjunto de números não repetidos *x* (1 <= x <= 100) e será terminada com um -1.
- Saída: A saída será os resultados de cada uma das *p* partidas. A informação de cada partida deverá iniciar com "PARTIDA #", em que # representa o índice da partida. Na linha seguinte, o programa deverá imprimir: "SEM VENCEDOR" caso a sequência de números "sorteados" da partida não seja suficiente para que haja vencedor (ou seja, a partida termine prematuramente); ou, "COM VENCEDOR", caso contrário. Nesse último caso, a lista das cartelas vencedoras deverá ser impressa (a ordem das tabelas vencedoras deverá ser a mesma da leitura inicial). Ver formato nos exemplos abaixo!

Você pode reutilizar as funções já implementadas na questão anterior. Adicionalmente, incremente o tipo *tCartela* para lidar com os requisitos desse problema. As funções abaixo devem ser adicionadas ao tipo *tCartela*:

- *MarcaCartela*: Essa função deve fazer a marcação de uma cartela com um número, ambos recebido como parâmetros e retornar a tabela marcada.
- *VenceuCartela*: Essa função deve retornar verdadeiro se uma cartela (recebida como parâmetro) tiver vencido a partida.
- *ResetaCartela*: Essa função deverá receber uma cartela, mudar o conteúdo dela para o estado inicial de quando ela foi lida (isto é, colocá-la como se a partida ainda não tivesse começado) e retorná-la.

Incremente também o tipo *tPartida* para lidar com os requisitos desse problema. A função abaixo (cabeçalho de livre escolha) deve ser adicionada ao tipo *tPartida*:

- *ResetaPartida*: Essa função deverá receber uma partida, mudar o conteúdo dela para o estado inicial de quando ela foi lida (isto é, colocá-la como se a partida ainda não tivesse começado) e retorná-la.
- *JogaPartida*: Essa função é responsável por jogar uma única partida. Ela assume que uma sequência de números a serem "sorteados" será passada pela entrada padrão e que a lista terminará com -1. A função deve consumir os números "sorteados" da entrada padrão, realizar as devidas marcações nas cartelas da partida (utilizando as funções apropriadas criadas anteriormente) e anunciar os

resultados da partida. A função deve garantir que todos os números sorteados para a partida em questão foram consumidos (inclusive o -1).

O aluno deverá utilizar a função *main* dada abaixo, ou seja, ela e seu conteúdo não poderão ser alterados:

```
int main(){
	tPartida partida;
	int qtdPartidas, i;

	partida = LeCartelasPartida();

	scanf ("%d", &qtdPartidas);
	for (i = 0; i < qtdPartidas; i++){
		if ( i!=0 ) printf("\n");
		printf("PARTIDA %d\n", i+1);
		partida = ResetaPartida(partida);
		JogaPartida(partida);
	}
	return 0;
}
```

- Exemplo de Entrada:

```
3
1 3
39 17 20 56 44 23 75 73 21
2 3
9 48 99 14 97 21 85 59 63
3 3
48 79 76 68 99 27 38 13 93
5
2 86 72 47 5 34 67 97 19 32 71 10 68 29 44 26 -1
9 48 99 14 97 21 85 59 -1
39 17 20 56 44 23 75 73 21 -1
1 9 14 21 59 63 85 97 99 13 27 38 76 79 93 48 68 30 -1
1 9 14 21 59 63 85 97 99 13 27 38 68 76 79 93 48 30 -1
```

- Exemplo de Saída:

```
PARTIDA 1
SEM VENCEDOR

PARTIDA 2
SEM VENCEDOR

PARTIDA 3
COM VENCEDOR
ID:1
039|056|075
017|044|073
020|023|021

PARTIDA 4
COM VENCEDOR
ID:2
009|014|085
048|097|059
099|021|063

PARTIDA 5
COM VENCEDOR
ID:2
009|014|085
048|097|059
099|021|063
ID:3
048|068|038
079|099|013
076|027|093
```