

You have been invited to participate on a Game Show! On the show, you are presented with a variety of prizes, each with an associated price. Your task is to add to your basket any combination of prizes whose prices add up exactly to a given total price target. If you do, you win all the prizes in your basket! If you accurately determine that there is no combination of prizes that add up to the exact price total target, you win all the prizes! Fortunately for you, you have a secret weapon: the backtracking algorithm!

Complete the backtracking algorithm by Implementing the following methods in the **gameshow.GameShow** class:

- `getBasketPriceTotal`
- `getSuccessors`
- `isValid`
- `isGoal`

Note that **HashSet** has a copy constructor you can use to make a copy of an existing **Set**:

```
Set<Integer> integerSet = new HashSet<>();  
Set<Integer> newIntegerSet = new HashSet<>(integerSet)
```

Sample runs have been provided in the **GameShow.main()** method. If a combination of prizes is found that adds up to the exact price total target, you will see output similar to:

```
Basket:{700, [MOUSE:10, THROW_RUG:20, BIKE:150, RICE_A_RONI:1,  
KEURIG:200, DETERGENT:12, TV:300, TURTLE_WAX:7]}, Price Target:  
700, Prizes: [TOASTER_OVEN:30, GUM:2, VR_HEADSET:300, LUGGAGE:80,  
CHAIR:15, MICROWAVE:80, LAPTOP:700]
```

Otherwise, if there is no solution, you will see:

```
No solution found
```