

Title

A stack of plates

Problem Description

At buffet restaurants there is often a stack of plates that diners can pick from. They can only pick the last plate that was placed on the stack, they cannot take a plate if the stack is empty, and servers may add more plates as long as the stack isn't too high.

Initialize an array of strings of length 10. In a loop, prompt the user to either add some number of plates or take a plate.

If the user chooses to add plates, ask them how many they want to add, and add them to the array. Do this by asking them what colour each plate is, and adding that colour to the end of the array. Users cannot add more plates than the array can store, but you may choose how to handle the case where the number of plates added exceeds the number of spots available.

If the user chooses to remove a plate, they may do so as long as there are plates available. Print the colour of the plate to the screen so the user knows what they got, and re-arrange the array appropriately. Remember, the user can only get the **last** plate from the array. They are not free to choose.

Interesting note:

This may seem like an unimportant problem, but in reality this is one way to implement a very important data structure known, unsurprisingly, as a stack. In the stack data structure, the last resource to be put in is the first resource to come out. This is useful, for example, to organize and allocate memory.

Testing

Assume that the first user enters three red plates, the second user takes a plate, and the third user enters two green plates. The table below shows what the array will look like after initialized (top row), the plates are entered (second row), the plate is removed (third row), and the two green plates are entered (bottom row).

EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY
RED	RED	RED	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY
RED	RED	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY
RED	RED	GREEN	GREEN	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY	EMPTY

Time Target

*** less than 15 minutes

** 15-25 minutes

* greater than 25 minutes

Section

Arrays