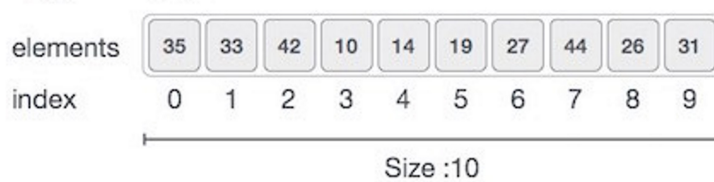


June1

30 May 2020 18:23

## Array and Multidimensional Array

### Array Representation



### Basic Operations

Following are the basic operations supported by an array.

- **Traverse** – print all the array elements one by one.
- **Insertion** – Adds an element at the given index.
- **Deletion** – Deletes an element at the given index.
- **Search** – Searches an element using the given index or by the value.
- **Update** – Updates an element at the given index.

Array is created in Python by importing array module to the python program.

```
from array import *  
  
arrayName = array(typecode, [Initializers])
```

**Note:** Python does not have built-in support for Arrays

#### # Question 1

Given an array  $A[0 \dots n - 1]$ , where each element of the array represents a vote in the election. Assume that each vote is given as an integer representing the ID of the chosen candidate. Give an algorithm for determining who wins the election.

#### # Question 2

Given an array  $A[]$  of  $n$  numbers and another number  $x$ , determines whether or not there exist two elements in  $A[]$  whose sum is exactly  $x$ .

#### # Question 3

Given a matrix of  $m \times n$  elements ( $m$  rows,  $n$  columns), return all elements of the matrix in spiral order.

#### # Question 4

Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).

The replacement must be in-place and use only constant extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.

1,2,3 → 1,3,2  
 3,2,1 → 1,2,3  
 1,1,5 → 1,5,1

TypeCode	C Type	Python Type	Min. size in bytes
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	Py_UNICODE	Unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'f'	float	float	4
'd'	double	float	8

Matrix:

```

  1 → 2 → 3 → 4
                    ↓
  5 → 6 → 7 → 8
  ↑           ↓   ↓
  9  10 ← 11  12
  ↑           ↓   ↓
 13 ← 14 ← 15 ← 16
  
```

Output:

1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10