**Review Session 2: Arrays**

## Question

What happens when you execute the following line of code?

int array[5]={10,20,30};

## Answer

array

```
array[0]
```

| 10 | 20 | 30 | 0 | 0 |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
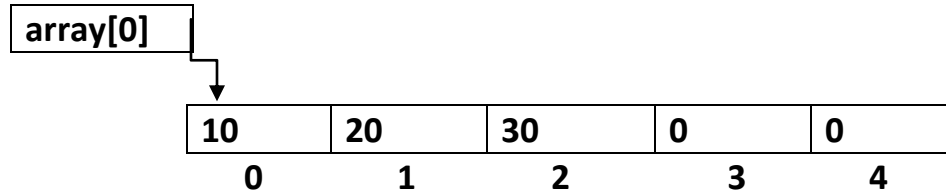
## Question

How can I pass an array to a function?
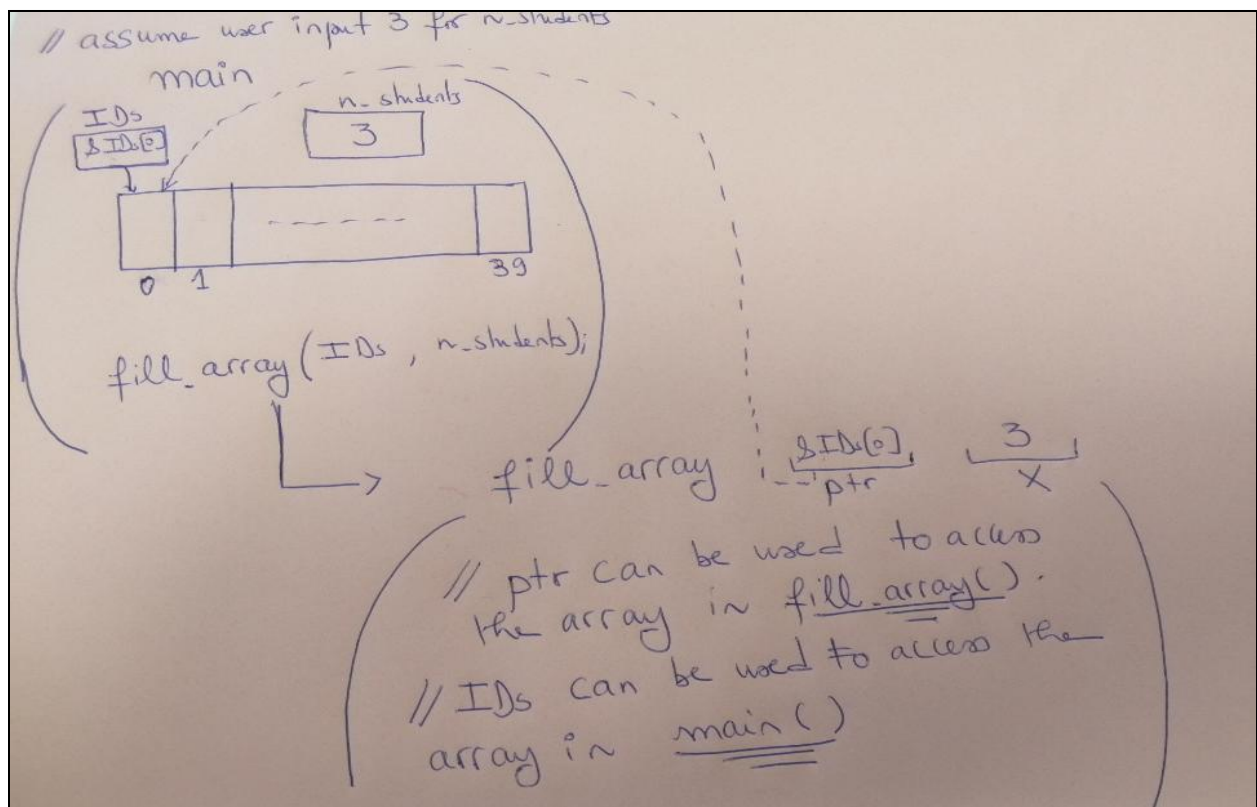
## Answer

**A rule to remember:** access to the first element in the array is enough to access the whole array

- This means that passing the address of the first element (ex: &array[0]) is enough to access the whole array
  - Passing the name of the array is also enough to access the whole array as the name of the array contains the address of the first element in the array!

Consider these figures: READ CAREFULLY THE COMMENTS

```c
1 #include<stdio.h>
2 void fill_array(int*,int);
3
4 main(){
5    int IDs[40]; //We don't know yet the number of students so predict the max 40
6    int n_students;
7
8    printf("Input the number of students:");
9    scanf("%d",&n_students);
10   fill_array(IDs,n_students);
11 }
12 void fill_array(int*ptr,int X){
13   int i;
14   for(i=0;i<X;i++)
15   {
16     printf("Input ID#%d:",i+1);
17     //We can not use IDs as it is a name seen in main() only!
18     scanf("%d",&ptr[i]);
19   }
20 }
```

```
// assume user input 3 for N_students
    main
IDs              n_students
$ID[0]              3

┌──┬──┬─────────┬──┐
│  │  │ - - - - │  │
└──┴──┴─────────┴──┘
0   1              39

fill_array (IDs , n_students);

    └──→    fill_array    $ID[0]       3
                          ---ptr        x

        // ptr can be used to access
           the array in fill_array().

        // IDs can be used to access the
           array in  main()
```

### Exercise 1

Using two arrays, IDs and Income (these two arrays will contain N clients IDs and their monthly incomes respectively) **declare, define and test** the following functions (you decide about the input and output arguments):

**fill_arrays():** A function that will be used to populate the arrays with some data

**display_arrays_content():** A function that will be used to display the arrays content in a **tabular format**

**search_income_client():** A function that will return the monthly income of a particular client (if exists!)

**sort_income():** A function that will sort clients using their monthly income in a descending order (you need to use bubble sort)

When you are done with implementing all these functions the best will be to provide the user with a menu to choose between: fill arrays, display arrays content, search for a client income, sort income descending, or quit!

☞ Logically the user can choose display, search, or sort only if the arrays are already filled!