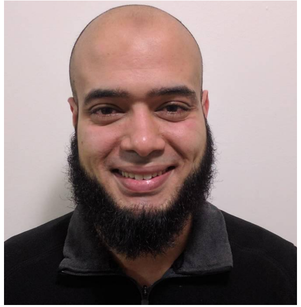
C++ Programming

1D Arrays

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

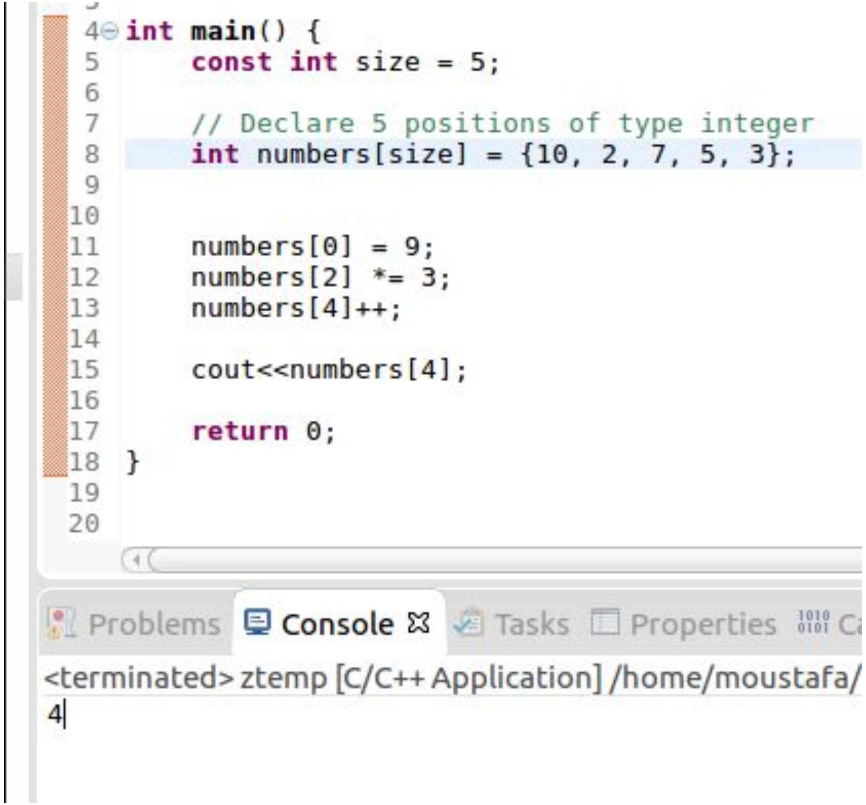
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Write a program that:

- That reads 1000 integers and print them reversed!
- That reads 1000 integers and find pairs of numbers with sum 12345?
- We can define 1000 variables! But this is a crazy idea!
- Programming languages introduce datatype **array** of **size K**
    - K variables defined in the memory (consecutively)
    - They all of **same data** type
- So now we create an array of size 1000
    - Then print them reversed!
    - That is all

Declare an array

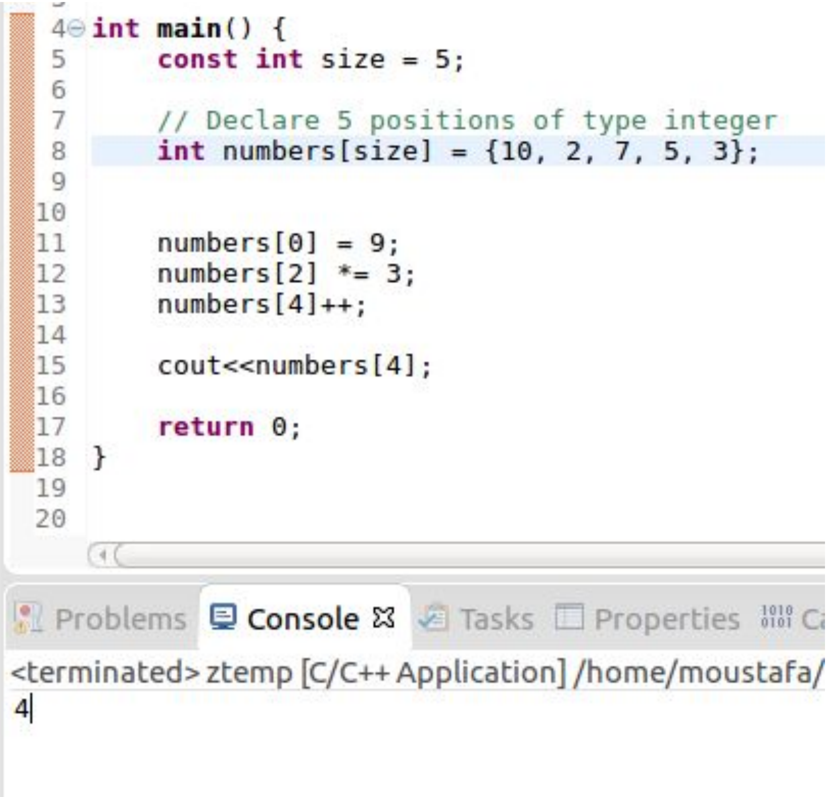


```
int main() {
    const int size = 5;

    // Declare 5 positions of type integer
    int numbers[size] = {10, 2, 7, 5, 3};


    numbers[0] = 9;
    numbers[2] *= 3;
    numbers[4]++;

    cout<<numbers[4];

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/
4
```

- Int number = 10;
- Int numbers**[5]**;
  - Create 5 numbers (variables)
    - You can’t change later!
  - Type integer
- numbers[i]
  - Access **ith** number
  - Completely like normal variable
  - We can read/output/change
- **Zero indexing**
  - numbers[0] first variable
  - numbers[size-1] last variable

Declare an array



```
 4 int main() {
 5     const int size = 5;
 6
 7     // Declare 5 positions of type integer
 8     int numbers[size] = {10, 2, 7, 5, 3};
 9
10
11     numbers[0] = 9;
12     numbers[2] *= 3;
13     numbers[4]++;
14
15     cout<<numbers[4];
16
17     return 0;
18 }
19
20
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/
4
```

- Line 8 declare the array

Index	0	1	2	3	4
numbers	10	2	7	5	3

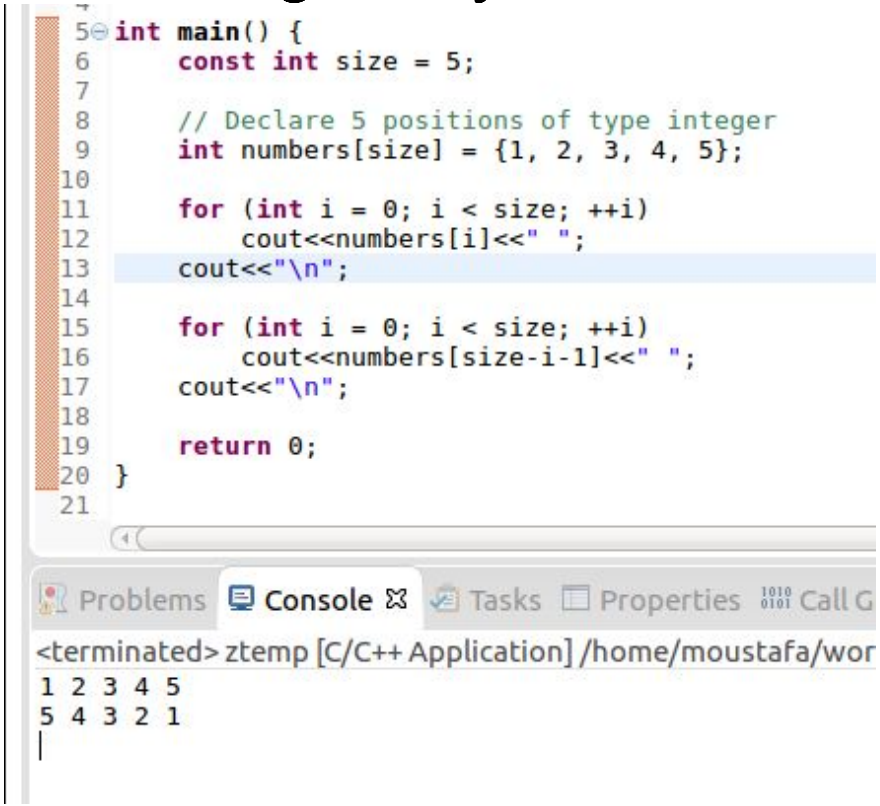
- Line 11 changes first number to 9

Index	**0**	1	2	3	4
numbers	**9**	2	7	5	3

- Line 12 and 13 also do changes

Index	**0**	1	**2**	3	**4**
numbers	**9**	2	**21**	5	**4**

Printing array forward and backward

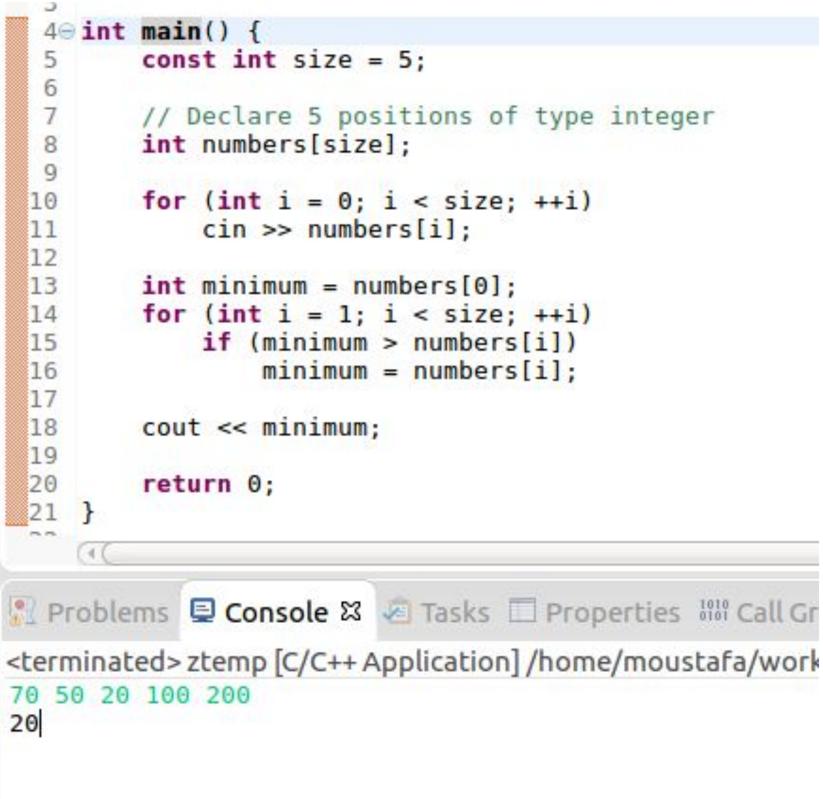


```
int main() {
    const int size = 5;

    // Declare 5 positions of type integer
    int numbers[size] = {1, 2, 3, 4, 5};

    for (int i = 0; i < size; ++i)
        cout<<numbers[i]<<" ";
    cout<<"\n";

    for (int i = 0; i < size; ++i)
        cout<<numbers[size-i-1]<<" ";
    cout<<"\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/wor
1 2 3 4 5
5 4 3 2 1
```

- Remember last element position is size-1
- Trace the backward
  - Index 4
  - Index 3
  - Index 2
  - Index 1
  - Index 0

Read 5 numbers in array - find minimum

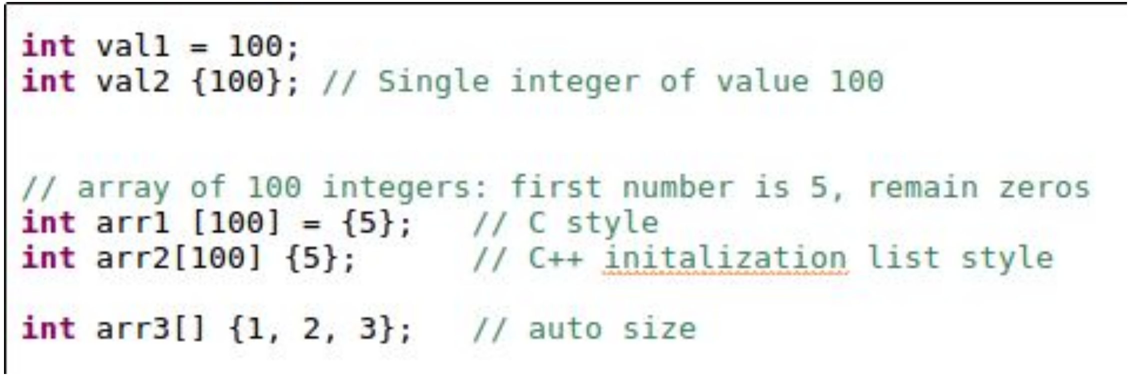


```
int main() {
    const int size = 5;

    // Declare 5 positions of type integer
    int numbers[size];

    for (int i = 0; i < size; ++i)
        cin >> numbers[i];

    int minimum = numbers[0];
    for (int i = 1; i < size; ++i)
        if (minimum > numbers[i])
            minimum = numbers[i];

    cout << minimum;

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/work
70 50 20 100 200
20
```

- Remember: Deal with each cell as a variable
  - read/write/assign
- Your turn
  - Write the codes and play with it

Initializations



```cpp
int val1 = 100;
int val2 {100}; // Single integer of value 100


// array of 100 integers: first number is 5, remain zeros
int arr1 [100] = {5};   // C style
int arr2[100] {5};      // C++ initalization list style

int arr3[] {1, 2, 3};   // auto size
```

Other Data types

- We can define array of other values
- **double** salary[100];
    - Array of 100 salaries
- **char** letters[300];
    - Array of 300 letters
- **string** names[200];
    - Array of 200 names

Run time error: Index out of boundary

- One of the most errors we do
- You access array with
  - Negative index
  - Index > its max value
- E.g. int arr[100];
- Don’t
  - arr[**100**]   ⇒ Only 0 to 99
  - arr[-10]
  - The program may **crash**
  - No one double checks the boundaries. You need to do by yourself

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”