Arrays

3

Arrays

- **Problem**

  read 100 integers and and print them reversed ?

- **Sol**

  you can define 100 variables! But this is huge!!

  So c++ introduce data type called Array

Arrays

- **What Is an Array?**

- An array is collection of elements of the same data type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier

Arrays

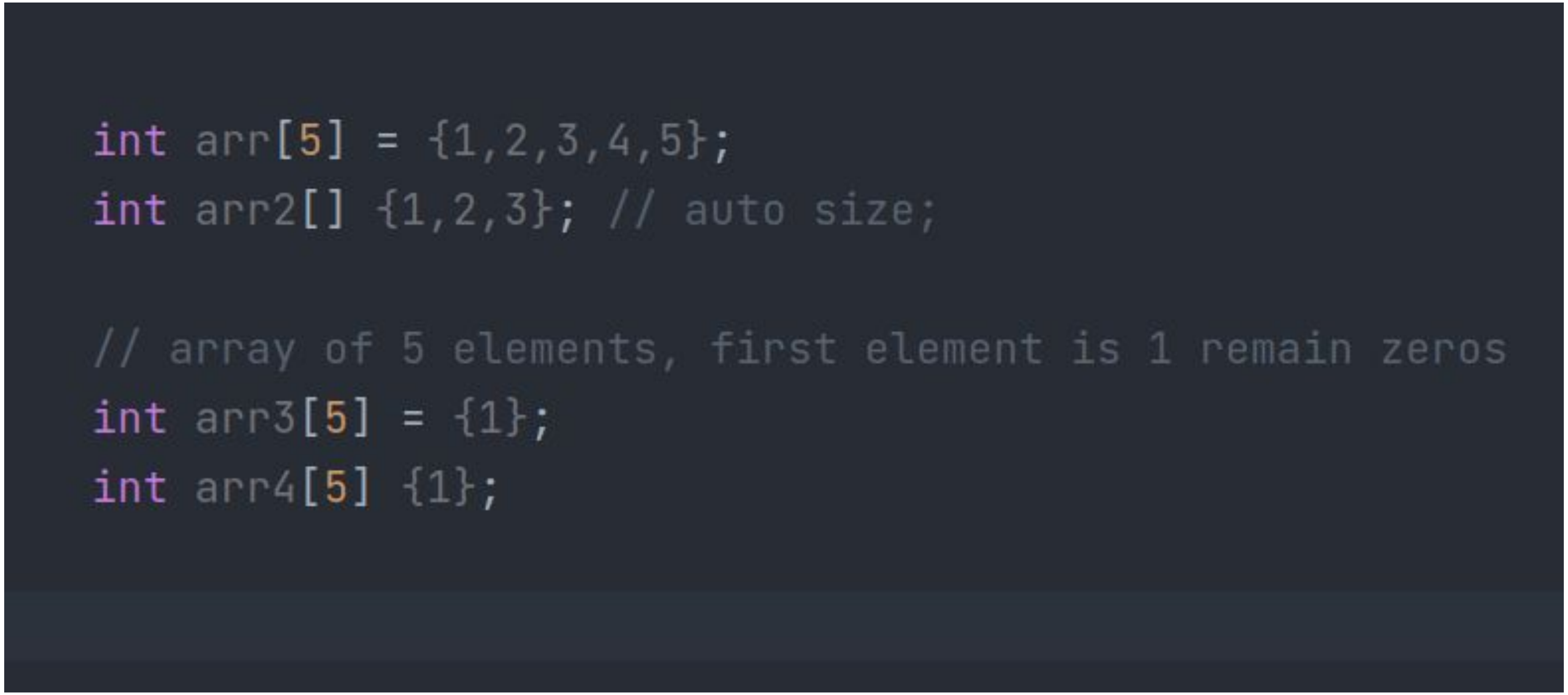
How to declare array?

to declare an array in C++ you must specify the following things

- The data type of the values which will be stored in the array
- The name of the array
- The dimensionality of the array
- The size of each dimension

Arrays

- **Initializations of an array**

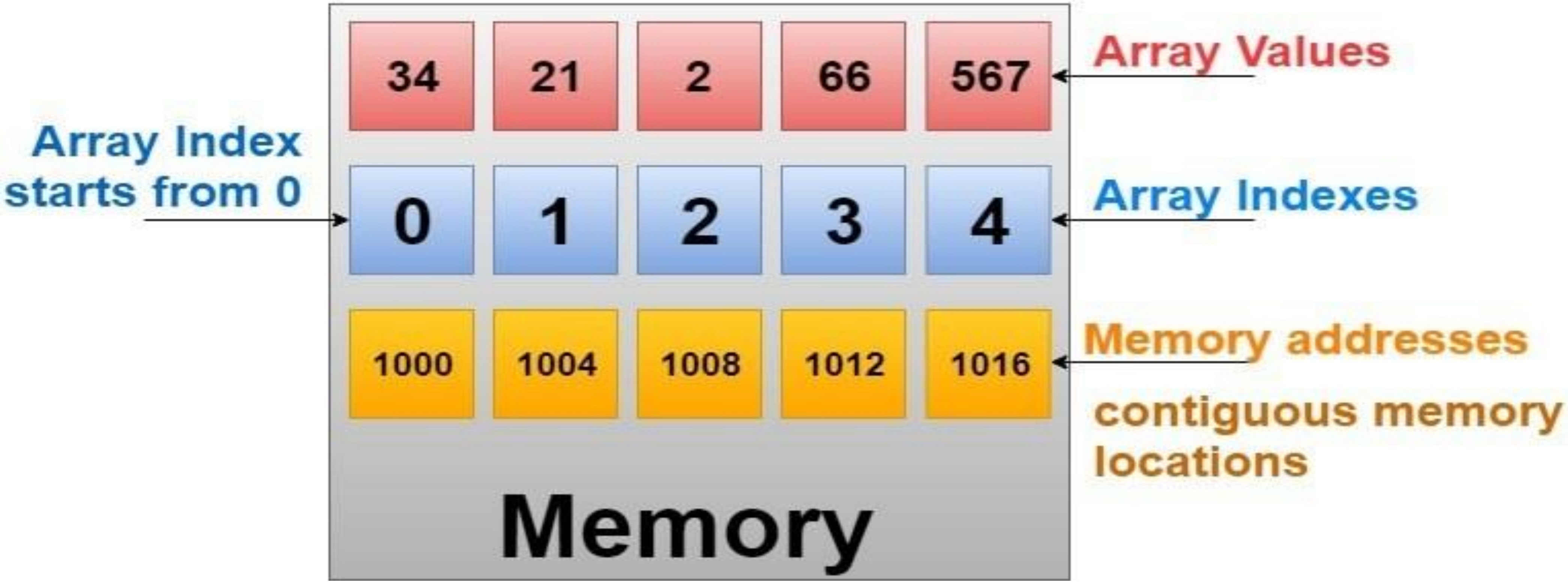


```
int arr[5] = {1,2,3,4,5};
int arr2[] {1,2,3}; // auto size;

// array of 5 elements, first element is 1 remain zeros
int arr3[5] = {1};
int arr4[5] {1};
```

Arrays

- **Initializations of an array**

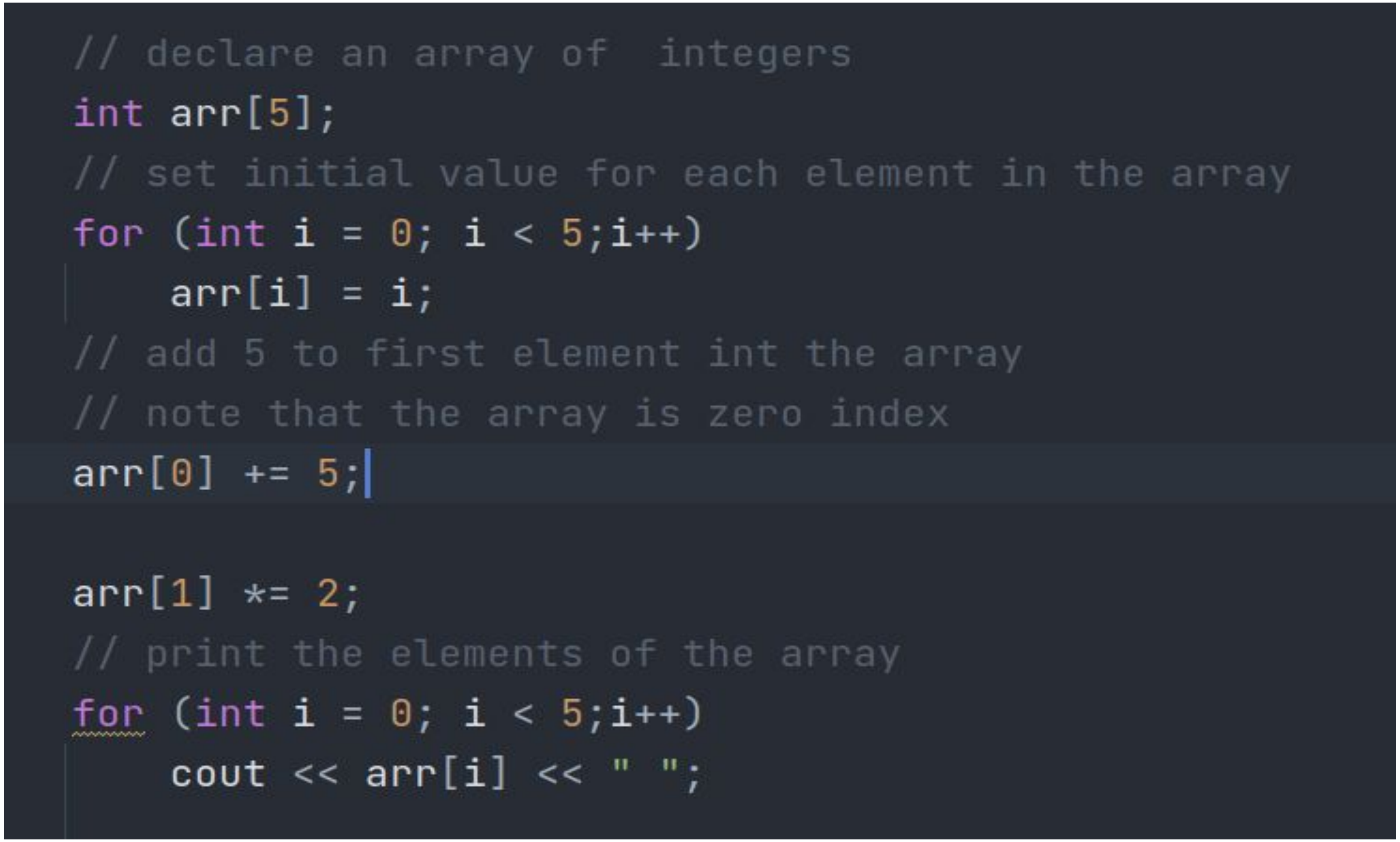
```
int x[5] = {34, 21, 2, 66, 567}
```



34	21	2	66	567
0	1	2	3	4
1000	1004	1008	1012	1016

Arrays

- **Accessing the values of an array**

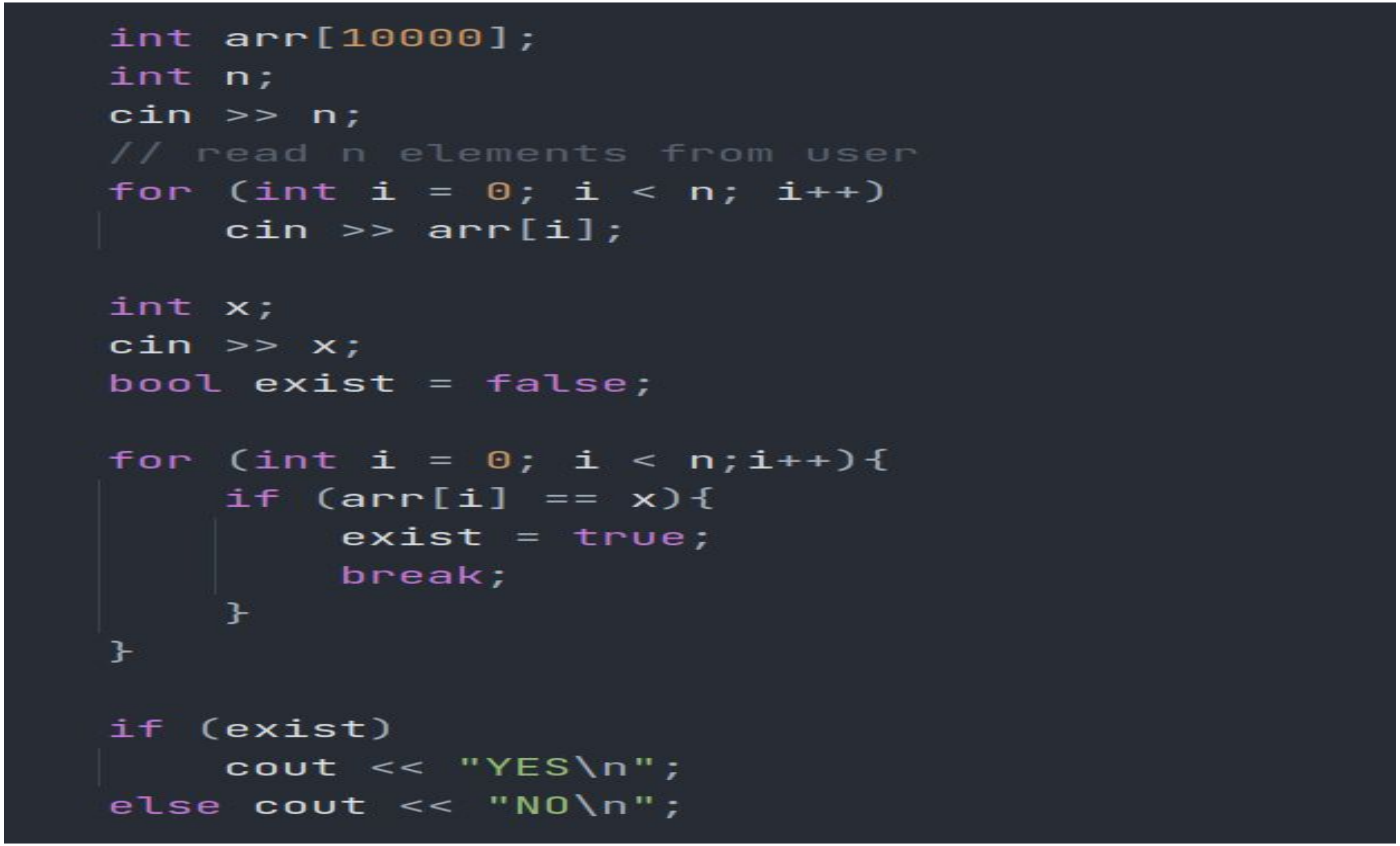


```
// declare an array of  integers
int arr[5];
// set initial value for each element in the array
for (int i = 0; i < 5;i++)
    arr[i] = i;
// add 5 to first element int the array
// note that the array is zero index
arr[0] += 5;

arr[1] *= 2;
// print the elements of the array
for (int i = 0; i < 5;i++)
    cout << arr[i] << " ";
```

Output: 5 2 2 3 4

Arrays

- Write a program to take array of size ***N*** from user and number ***X*** then check if ***X*** is exist in array or not



```
int arr[10000];
int n;
cin >> n;
// read n elements from user
for (int i = 0; i < n; i++)
    cin >> arr[i];

int x;
cin >> x;
bool exist = false;

for (int i = 0; i < n;i++){
    if (arr[i] == x){
        exist = true;
        break;
    }
}

if (exist)
    cout << "YES\n";
else cout << "NO\n";
```

Arrays

- **try by yourself**

- Write a program to take a **10 numbers** from user and print largest and smallest numbers.

- Write a program to take array of size ***N*** from user then if number even change its value to 0 otherwise to 1 then print the array.

Arrays

- ## Other data type

  - We can define array of other values

    double salary[100];

    Array of 100 salaries

    char letters[300];

    Array of 300 letters

    string names[200];

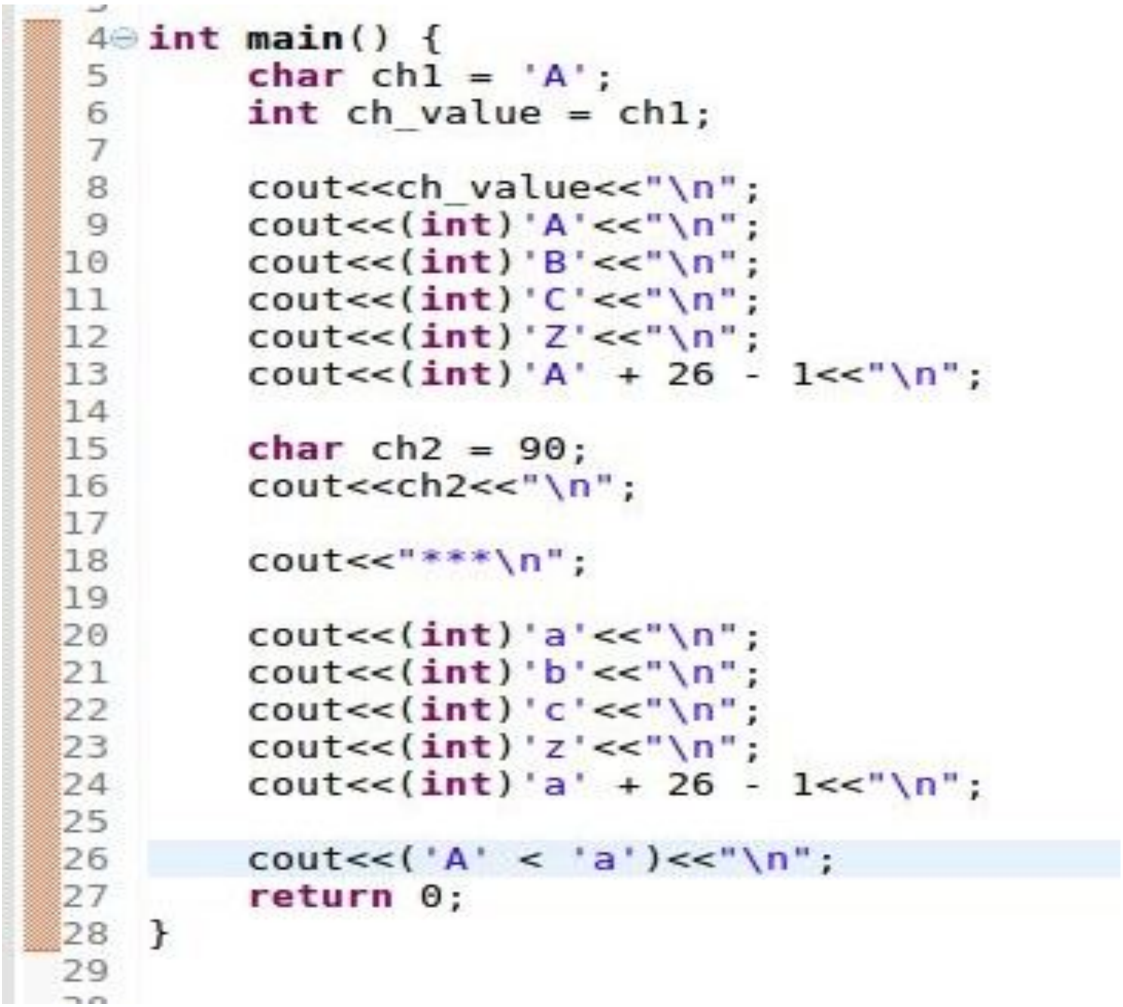
    Array of 200 names

Arrays

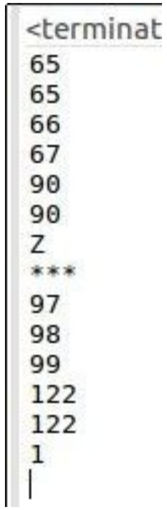
- **Run time error: index out of boundary**
  - One of the most errors we do
  - you can access array with
    - Negative index
    - Index > its max value
  - E.g. int arr[100];
    - Don’t
    - arr[**100**] ⇒ Only 0 to 99
    - arr[-10]
    - The program may **crash**
    - No one double checks the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

Arrays

- **Let’s refresh about characters**



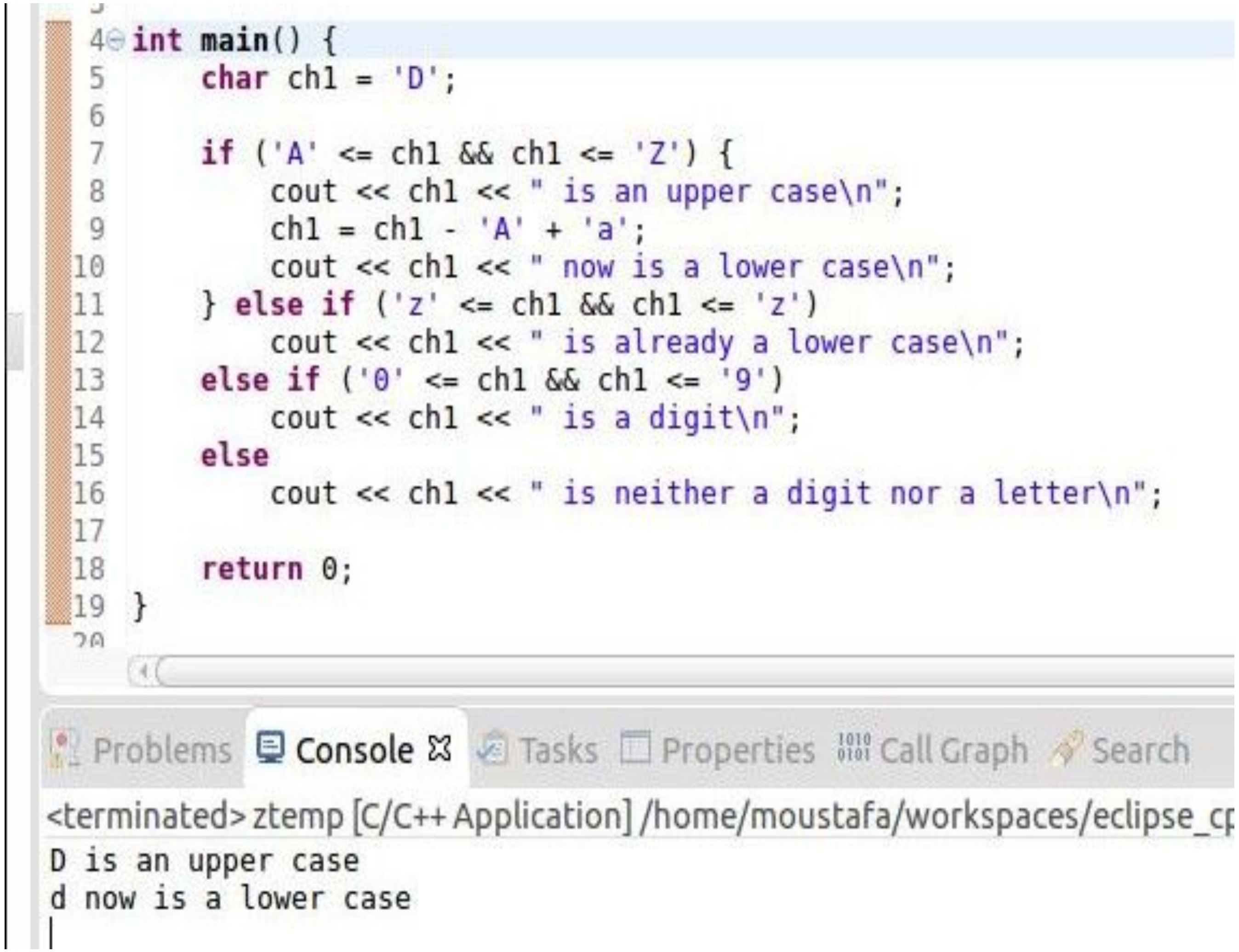
```
int main() {
	char ch1 = 'A';
	int ch_value = ch1;

	cout<<ch_value<<"\n";
	cout<<(int)'A'<<"\n";
	cout<<(int)'B'<<"\n";
	cout<<(int)'C'<<"\n";
	cout<<(int)'Z'<<"\n";
	cout<<(int)'A' + 26 - 1<<"\n";

	char ch2 = 90;
	cout<<ch2<<"\n";

	cout<<"***\n";

	cout<<(int)'a'<<"\n";
	cout<<(int)'b'<<"\n";
	cout<<(int)'c'<<"\n";
	cout<<(int)'z'<<"\n";
	cout<<(int)'a' + 26 - 1<<"\n";

	cout<<('A' < 'a')<<"\n";
	return 0;
}
```



```
<terminat
65
65
66
67
90
90
Z
***
97
98
99
122
122
1
```

Arrays

● Check and Convert Chars



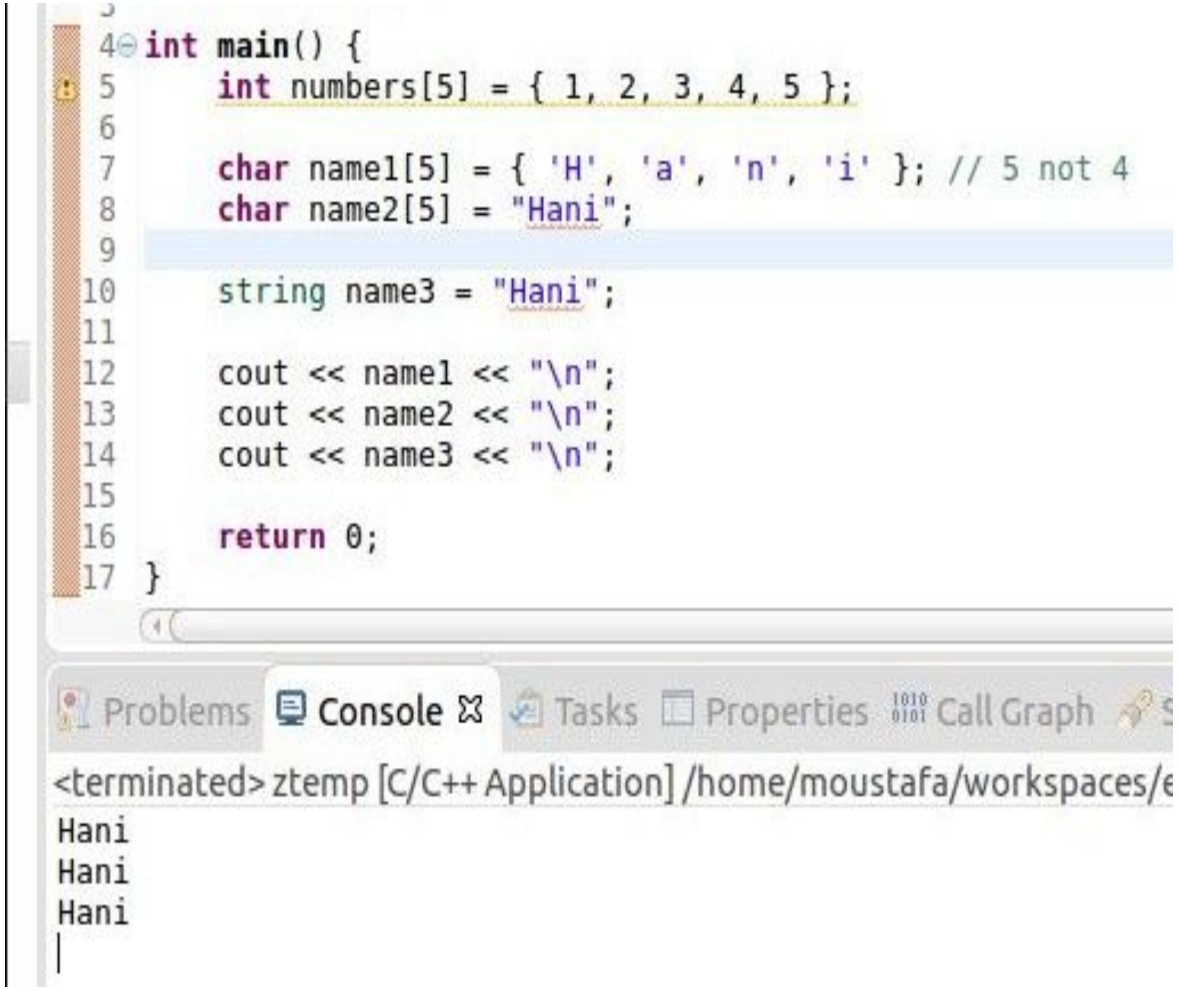
```
int main() {
    char ch1 = 'D';

    if ('A' <= ch1 && ch1 <= 'Z') {
        cout << ch1 << " is an upper case\n";
        ch1 = ch1 - 'A' + 'a';
        cout << ch1 << " now is a lower case\n";
    } else if ('z' <= ch1 && ch1 <= 'z')
        cout << ch1 << " is already a lower case\n";
    else if ('0' <= ch1 && ch1 <= '9')
        cout << ch1 << " is a digit\n";
    else
        cout << ch1 << " is neither a digit nor a letter\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/eclipse_cp
D is an upper case
d now is a lower case
```

- Always remember they are just numbers internally
- If we have letter ‘A’
  - ○ Subtract ‘A’
  - ○ Now this is zero
  - ○ Now add ‘a’
  - ○ This shifts to ‘a’
  - ○ And so on
  - ○ If ‘D’ ⇒ ‘D’ - ’A’ = 3

Arrays

- **Let’s create char array**



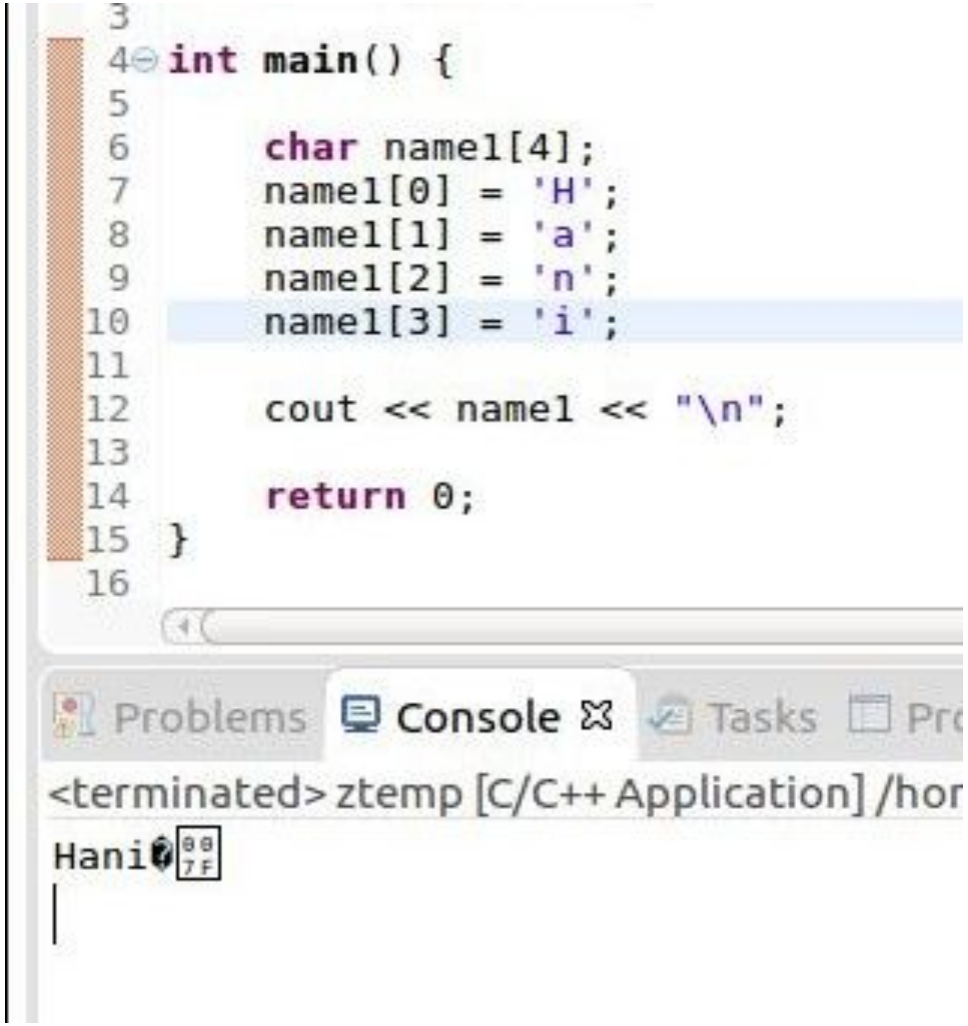
```
int main() {
	int numbers[5] = { 1, 2, 3, 4, 5 };

	char name1[5] = { 'H', 'a', 'n', 'i' }; // 5 not 4
	char name2[5] = "Hani";

	string name3 = "Hani";

	cout << name1 << "\n";
	cout << name2 << "\n";
	cout << name3 << "\n";

	return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /home/moustafa/workspaces/e
Hani
Hani
Hani
```

- We can create array of integers or doubles!
- Let’s create array of chars
- Length must be 1 + intended length
- Usually, you will use string as easier way
  - Internally has char array

Arrays

- **the null char**



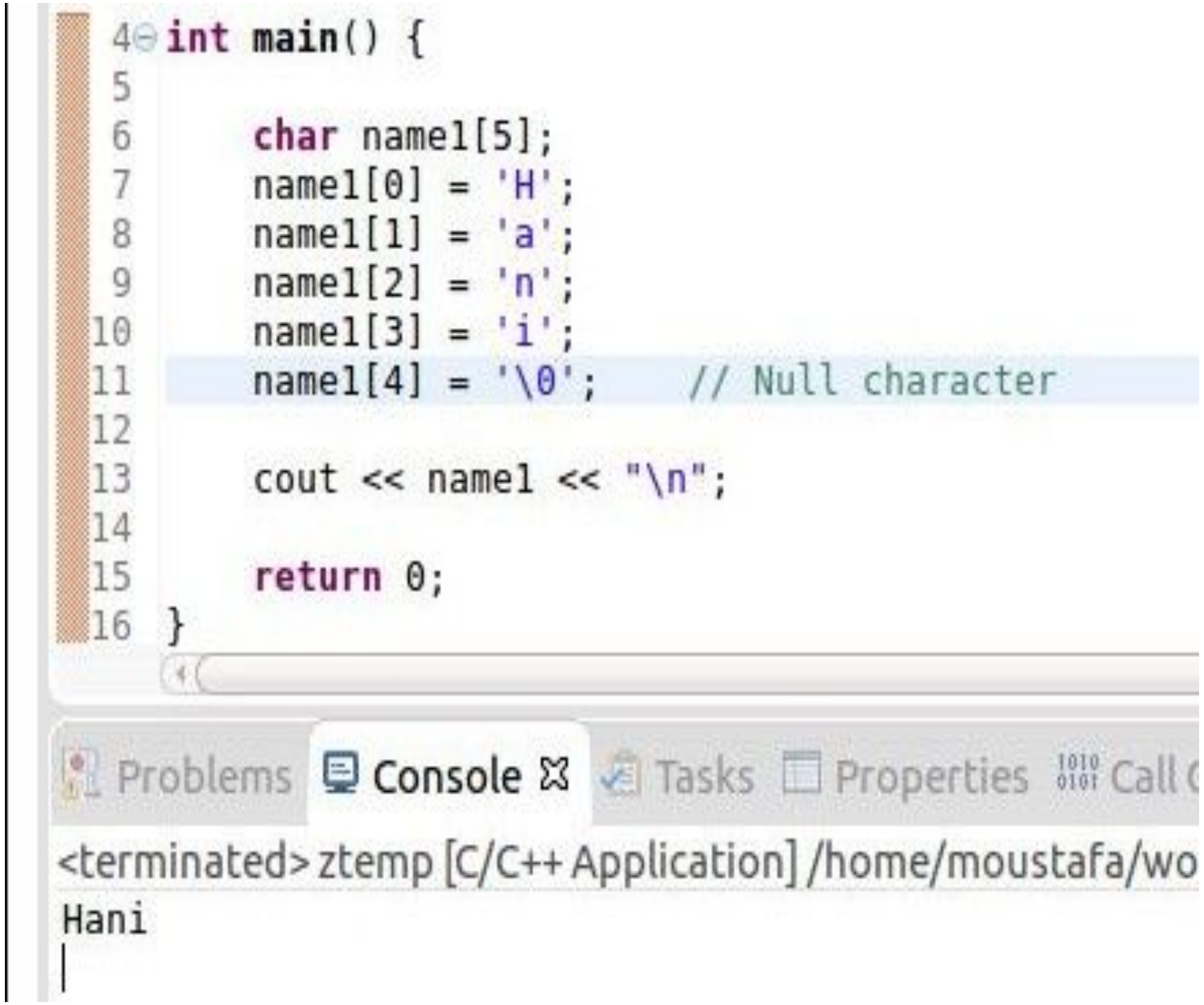
```
int main() {

    char name1[4];
    name1[0] = 'H';
    name1[1] = 'a';
    name1[2] = 'n';
    name1[3] = 'i';

    cout << name1 << "\n";

    return 0;
}
```

```
<terminated> ztemp [C/C++ Application] /hor
Hani��
```

- For internal reasons, C++ wanna you tell when the string ends (for easy print)
- We add extra char to do so
- On left, wrong way

Arrays

- **the null char**



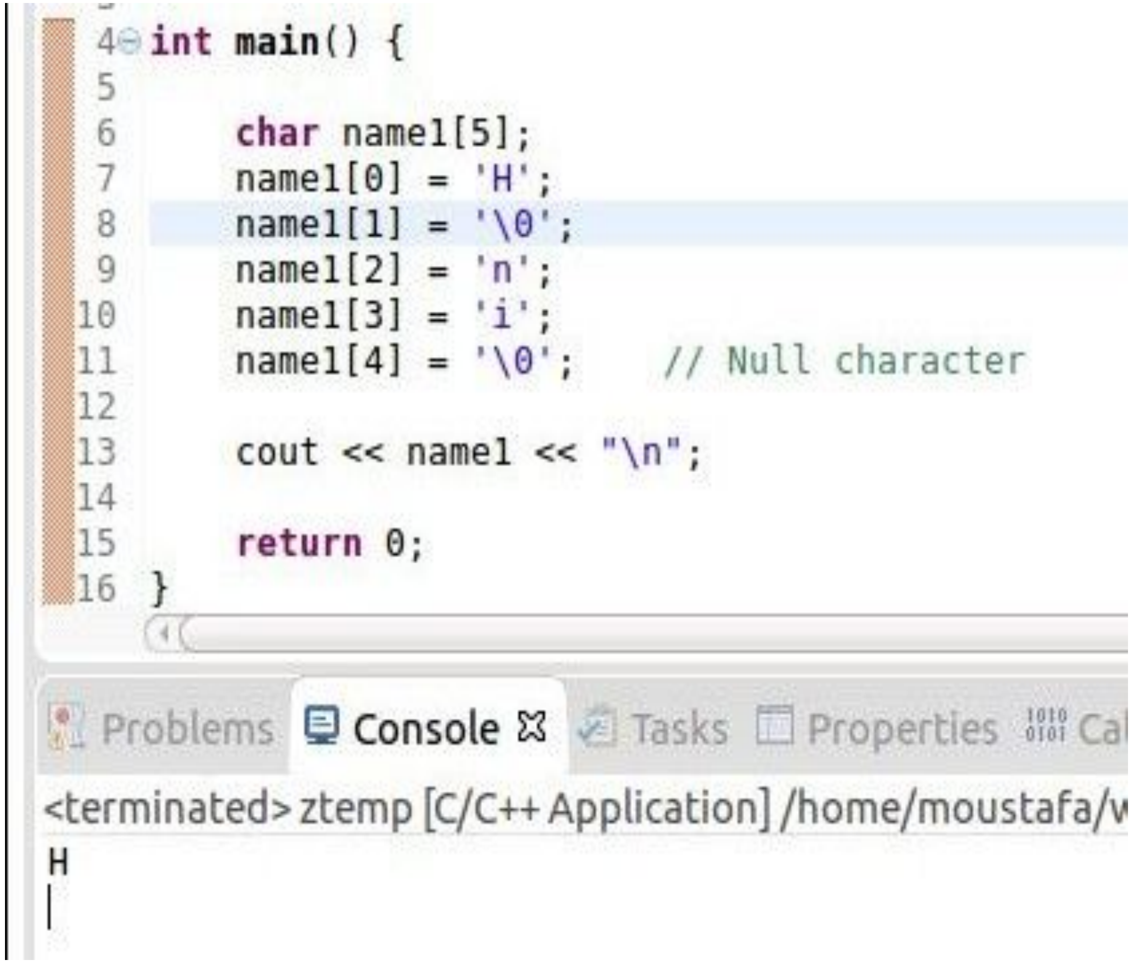
```
int main() {

    char name1[5];
    name1[0] = 'H';
    name1[1] = 'a';
    name1[2] = 'n';
    name1[3] = 'i';
    name1[4] = '\0';     // Null character

    cout << name1 << "\n";

    return 0;
}
```

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

- The right way
  - 1) 1+size (4 here)
  - 2) Null char: \0

Arrays

- **the null char**



```
int main() {

    char name1[5];
    name1[0] = 'H';
    name1[1] = '\0';
    name1[2] = 'n';
    name1[3] = 'i';
    name1[4] = '\0';     // Null character

    cout << name1 << "\n";

    return 0;
}
```

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

- Print stops once sees the null char
  - Letters after it won’t be printed

Arrays

- **try by yourself**

- Write a program to take array of char of size *N* from user then convert every char that are upper to lower and every char that are lower to upper

- Write a program to take array of char of size N form the user then convert every char in an even index into upper char and every char in the odd index into lower char

Arrays

- **2D Arrays Motivation**

- Write a program that reads grades for students
  - 100 students
  - 20 subjects
- How can we code that?
  - Create 20 arrays grade1[100], grade2[100], …..grade20[100];
  - So impractical!
- Let’s visualize the data

Arrays

- **Grades visualization: 7 students x 4 subjects**

	Math	Science	History	Arts
Mostafa	50	33	40	30
Asmaa	35	50	44	17
Belal	30	35	50	37
Ziad	50	35	44	22
Safa	50	44	50	30
Ashraf	50	36	18	50
Mona	37	30	47	16

- This is called a matrix/table
  - The blue numbers
- 7 rows
  - Row 0, 1, 2, … 6
  - Row 0 for mostafa
  - Row 6 for mona
- 4 Columns
  - Column 0, 1, 2, 3
  - Column 0 for Math
- Value of table: row 6, col 2
  - 47 (Mona & History)
  - Notation: [6][2]

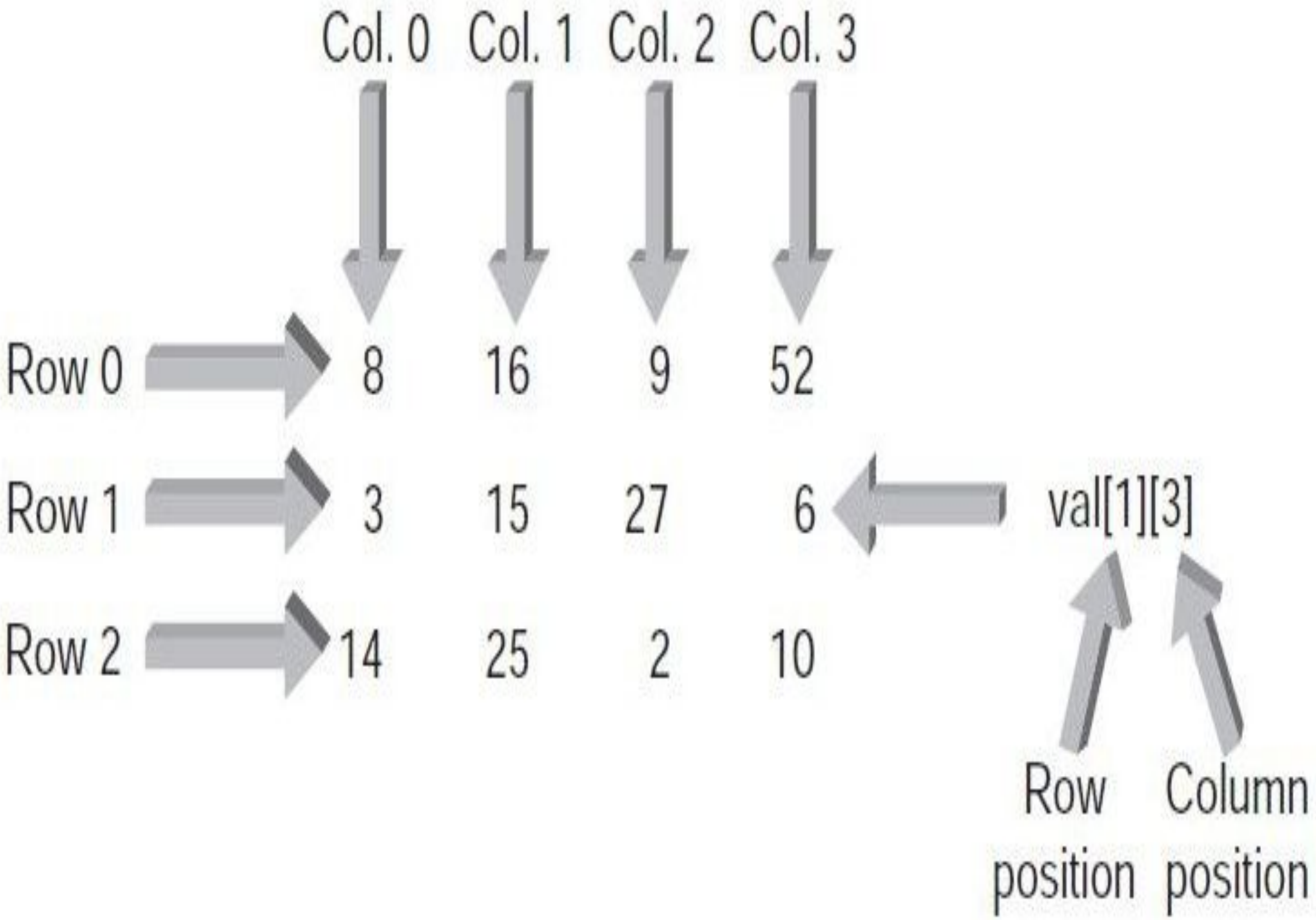
Arrays

- 2D Arrays

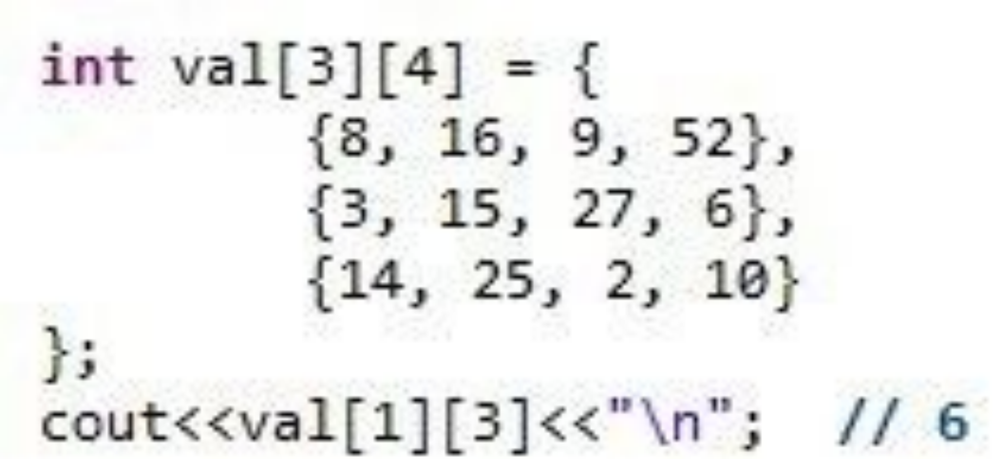
- C++ saves our time by using 2D arrays
  - 2D = Table: rows x columns
- Same rules as 1D Arrays
- We create it as
  - double grades[7][4];
    - For 7 rows and 4 columns
  - To access in 2D arrays:
    - grades[6][2]

Arrays

- 2D Arrays Visualization



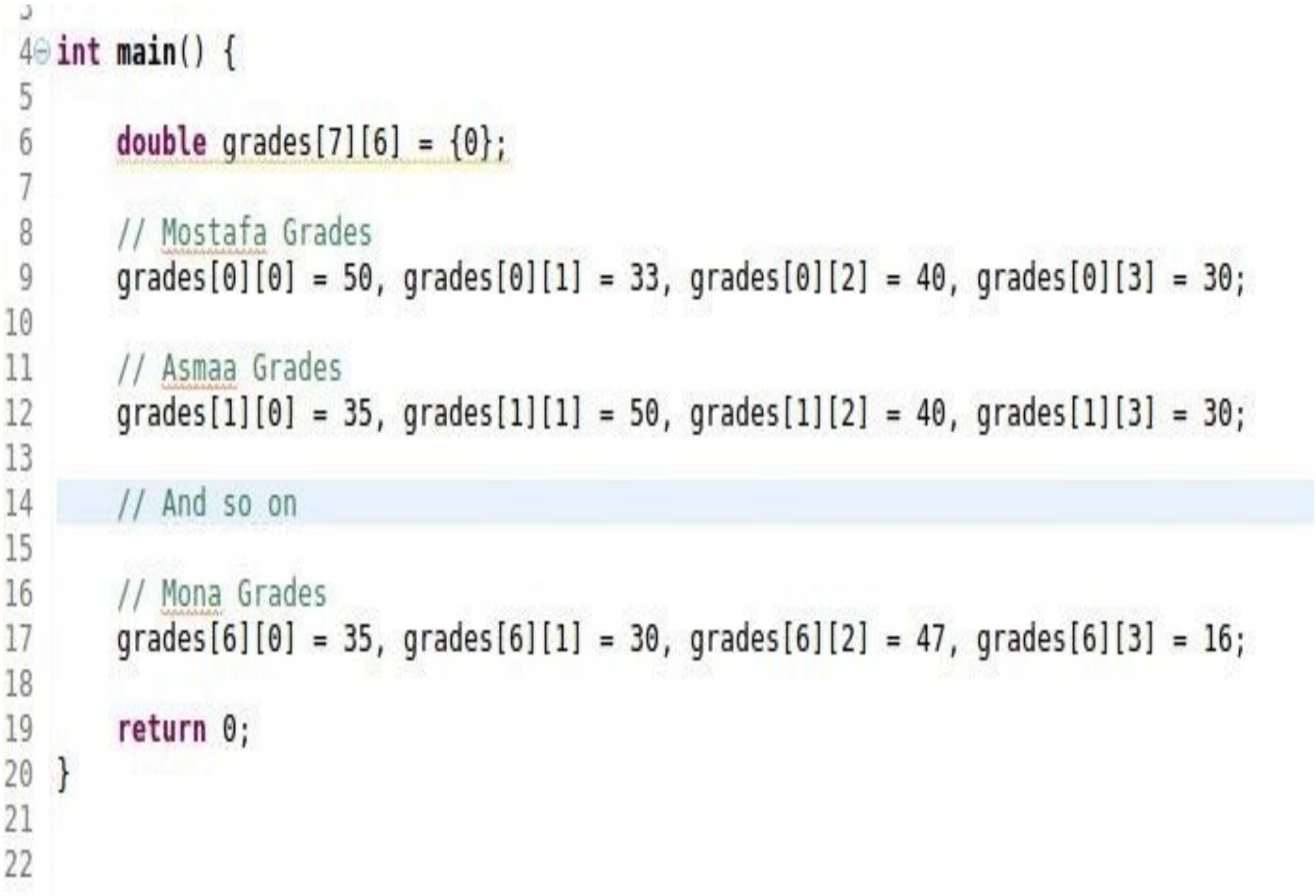
	Col. 0	Col. 1	Col. 2	Col. 3
Row 0	8	16	9	52
Row 1	3	15	27	6
Row 2	14	25	2	10



```
int val[3][4] = {
                {8, 16, 9, 52},
                {3, 15, 27, 6},
                {14, 25, 2, 10}
};
cout<<val[1][3]<<"\n";  // 6
```

Arrays

- Let’s put the values

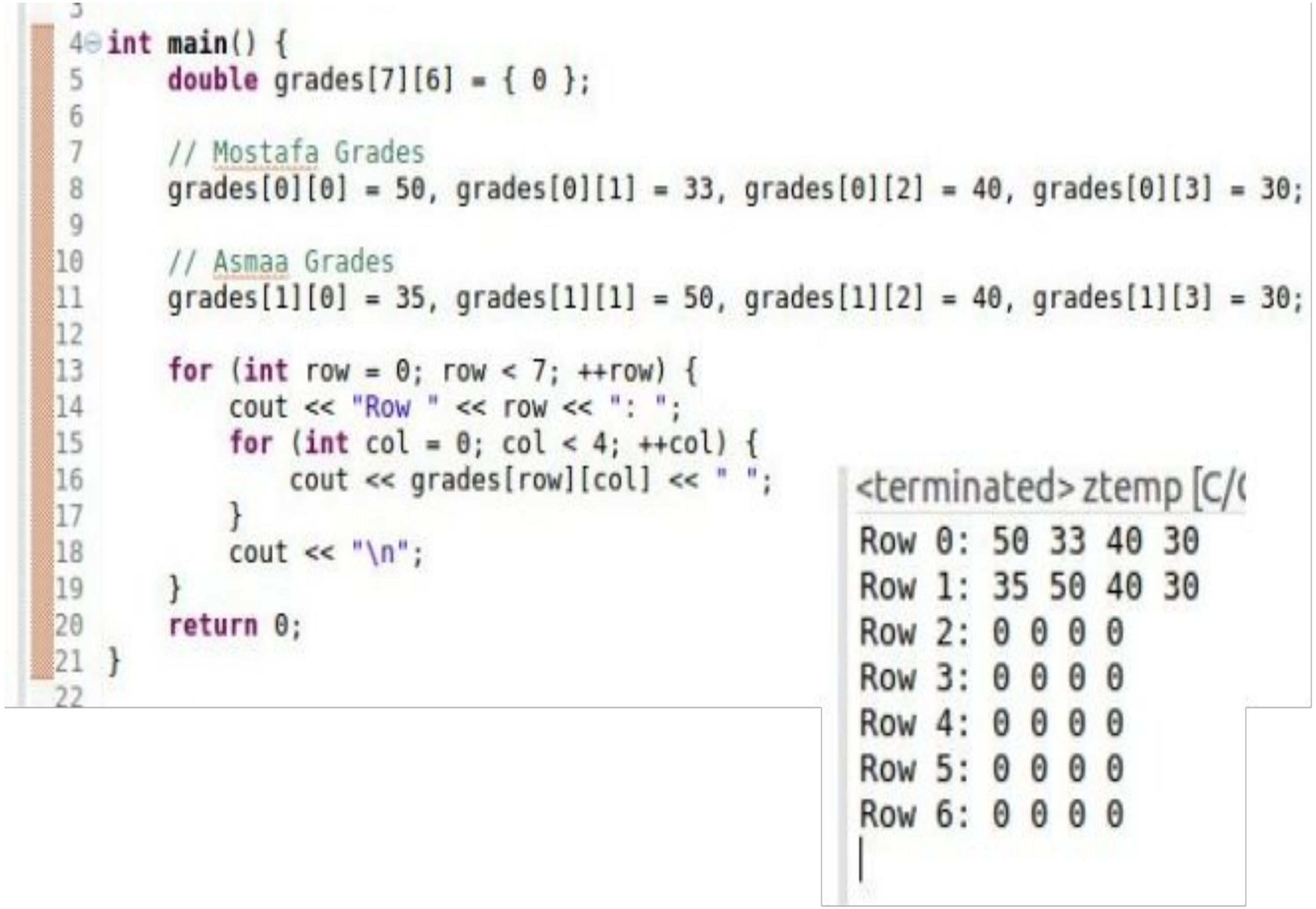


```
int main() {

    double grades[7][6] = {0};

    // Mostafa Grades
    grades[0][0] = 50, grades[0][1] = 33, grades[0][2] = 40, grades[0][3] = 30;

    // Asmaa Grades
    grades[1][0] = 35, grades[1][1] = 50, grades[1][2] = 40, grades[1][3] = 30;

    // And so on

    // Mona Grades
    grades[6][0] = 35, grades[6][1] = 30, grades[6][2] = 47, grades[6][3] = 16;

    return 0;
}
```

- Notice
- All mostafa data has grades[0]
- All Asmaa data has grades[1]
- All mona data has grades[6]
- Notice all inces
  - 0-6 for rows
  - 0-3 for columns

Arrays

- Let’s print it



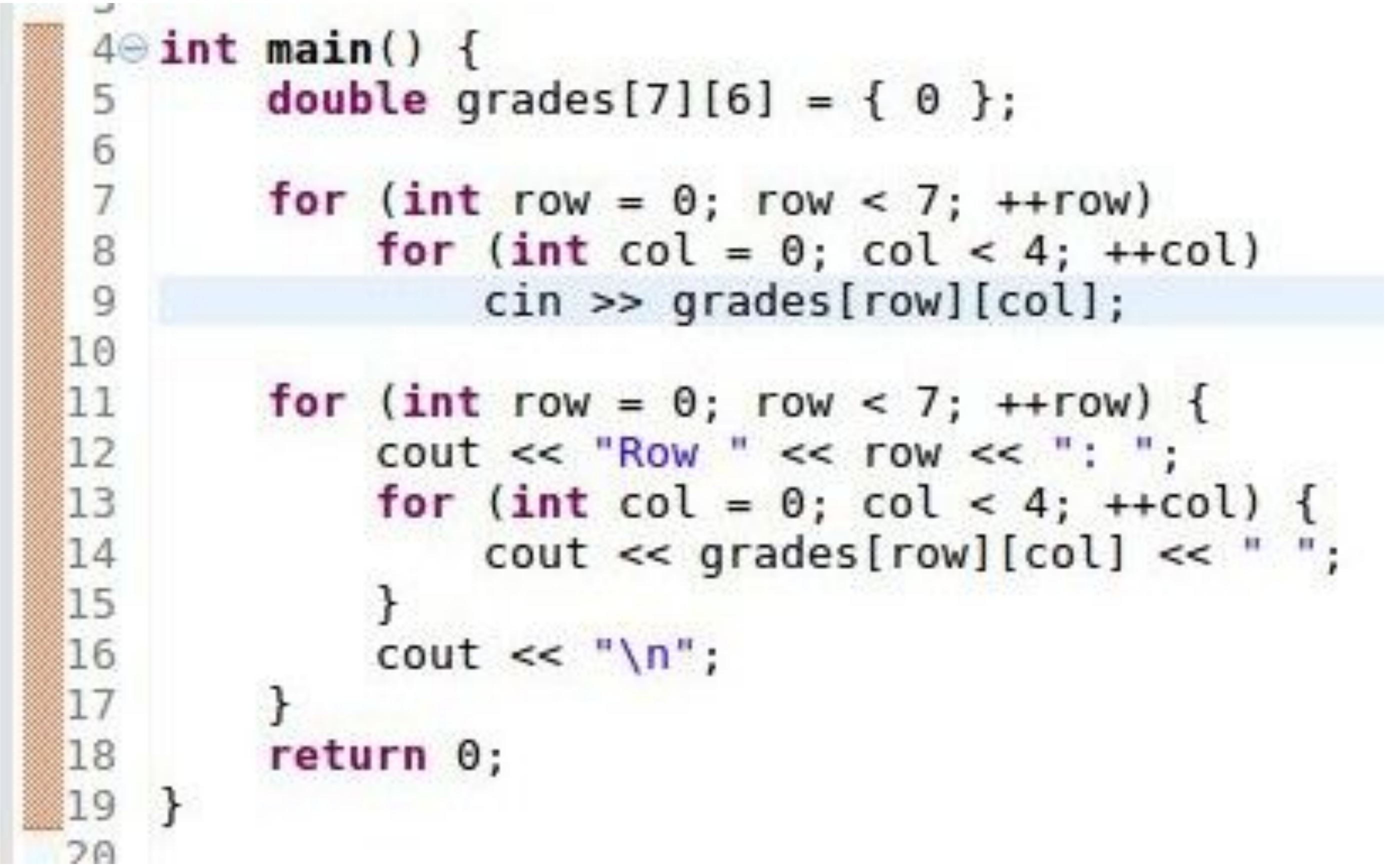
```
int main() {
    double grades[7][6] = { 0 };

    // Mostafa Grades
    grades[0][0] = 50, grades[0][1] = 33, grades[0][2] = 40, grades[0][3] = 30;

    // Asmaa Grades
    grades[1][0] = 35, grades[1][1] = 50, grades[1][2] = 40, grades[1][3] = 30;

    for (int row = 0; row < 7; ++row) {
        cout << "Row " << row << ": ";
        for (int col = 0; col < 4; ++col) {
            cout << grades[row][col] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

```
<terminated> ztemp [C/(
Row 0: 50 33 40 30
Row 1: 35 50 40 30
Row 2: 0 0 0 0
Row 3: 0 0 0 0
Row 4: 0 0 0 0
Row 5: 0 0 0 0
Row 6: 0 0 0 0
```

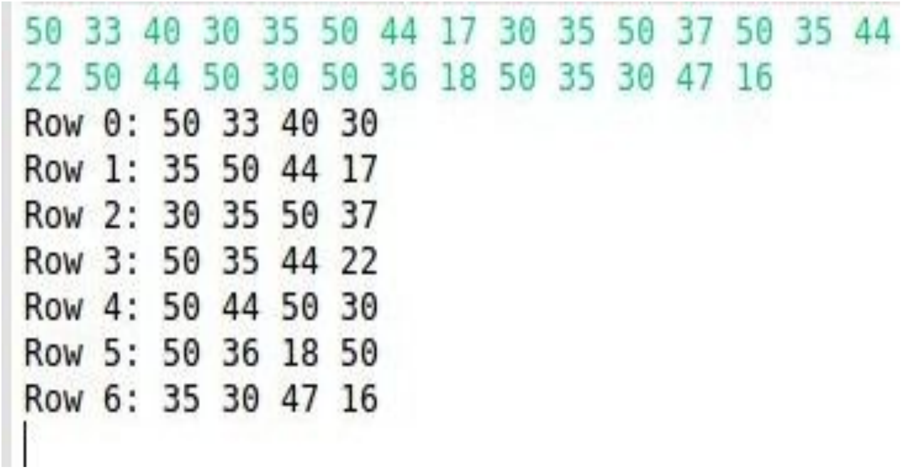
- To print
  - Loop over every row
  - Then for this row
    - Loop on its columns
- We will loop this way typically
- We can also loop on
- columns then loop on rows

Arrays

- Easier: Let’s read then print!



```
int main() {
    double grades[7][6] = { 0 };

    for (int row = 0; row < 7; ++row)
        for (int col = 0; col < 4; ++col)
            cin >> grades[row][col];

    for (int row = 0; row < 7; ++row) {
        cout << "Row " << row << ": ";
        for (int col = 0; col < 4; ++col) {
            cout << grades[row][col] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```



```
50 33 40 30 35 50 44 17 30 35 50 37 50 35 44
22 50 44 50 30 50 36 18 50 35 30 47 16
Row 0: 50 33 40 30
Row 1: 35 50 44 17
Row 2: 30 35 50 37
Row 3: 50 35 44 22
Row 4: 50 44 50 30
Row 5: 50 36 18 50
Row 6: 35 30 47 16
```

Arrays

- **try to solve**

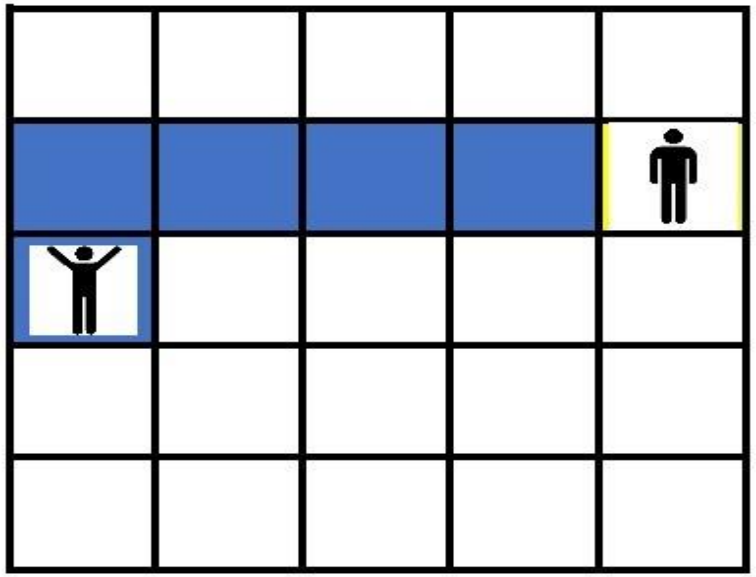
- Write a program to take an array 2D from user and print largest and smallest number in each row

- Write a program that takes 2D array of integers of size (N x N) from the user and print the summation of even numbers in each row and the summation of odd numbers in each row

Arrays

- **try to solve**

- You and your friend are lost in 2D array and you want to know the minimum distance between yours, you know your index (x1, y1) and your friend`s index (x2, y2), what is distance between yours ?



References

- Mohamed Al-Desoqi
  https://www.youtube.com/watch?v=UjltuSZ4plw

- Mostafa Saad
  https://www.youtube.com/watch?v=0HT2-2qD654

  https://youtu.be/-GxY9NCG9Bw

THANK YOU