INDEX ORDER

Does the order matter?

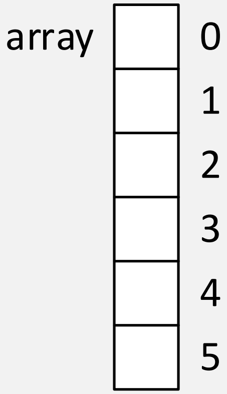
Delroy A. Brinkerhoff

ARRAY DEFINITION AND MEMORY ALLOCATION

```
int array[2][3];

int array[3][2];
```

rows × cols = cols × rows



REASONS FOR ADOPTING [ROWS][COLS]

- Accepted practice
- Similar usage to math: $x_{1,2}$
- Matches Java
- C++ sometimes represents text as an array of strings
- Printing arrays to the console
- Initializing arrays: int array[3][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
- Extracting rows from an array: array[2];

```
1  2  3  4
5  6  7  8
9 10 11 12
```

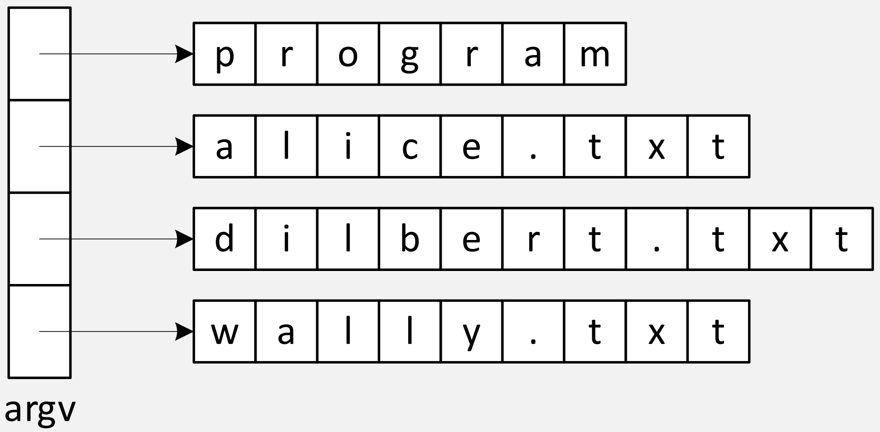
PROGRAMMING EXAMPLE

```
void print_row(int* row, int size)
{
    for (int i = 0; i < size; i++)
        cout << row[i] << endl;
}

int main()
{
    int array[3][4] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
    print_row(array[2], sizeof(array[2]) / sizeof(int));
    return 0;
}
```

1	2	3	4
5	6	7	8
9	10	11	12

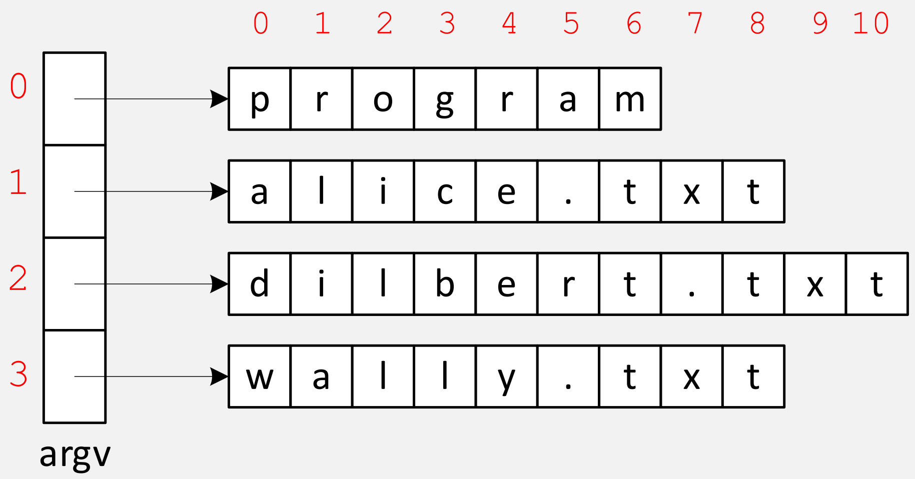
CHARACTER ACCESS: ARRAY OF STRINGS



p	r	o	g	r	a	m				
a	l	i	c	e	.	t	x	t		
d	i	l	b	e	r	t	.	t	x	t
w	a	l	l	y	.	t	x	t		

- Command line arguments
  - Come from the operating system
  - Are an array of strings
  - Individual characters are accessed with two indexes: [row][col]

CHARACTER ACCESS:
ARRAY OF STRINGS



	0	1	2	3	4	5	6	7	8	9	10
0	p	r	o	g	r	a	m				
1	a	l	i	c	e	.	t	x	t		
2	d	i	l	b	e	r	t	.	t	x	t
3	w	a	l	l	y	.	t	x	t		

- Command line arguments
  - Come from the operating system
  - Are an array of strings
  - Individual characters are accessed with two indexes: [row][col]
  - `argv[2][5] is 'r'`