

Lab9: Pointers and Arrays

Yasin Zamani

Dian-Lun Lin (Luan)

Department of Electrical and Computer Engineering

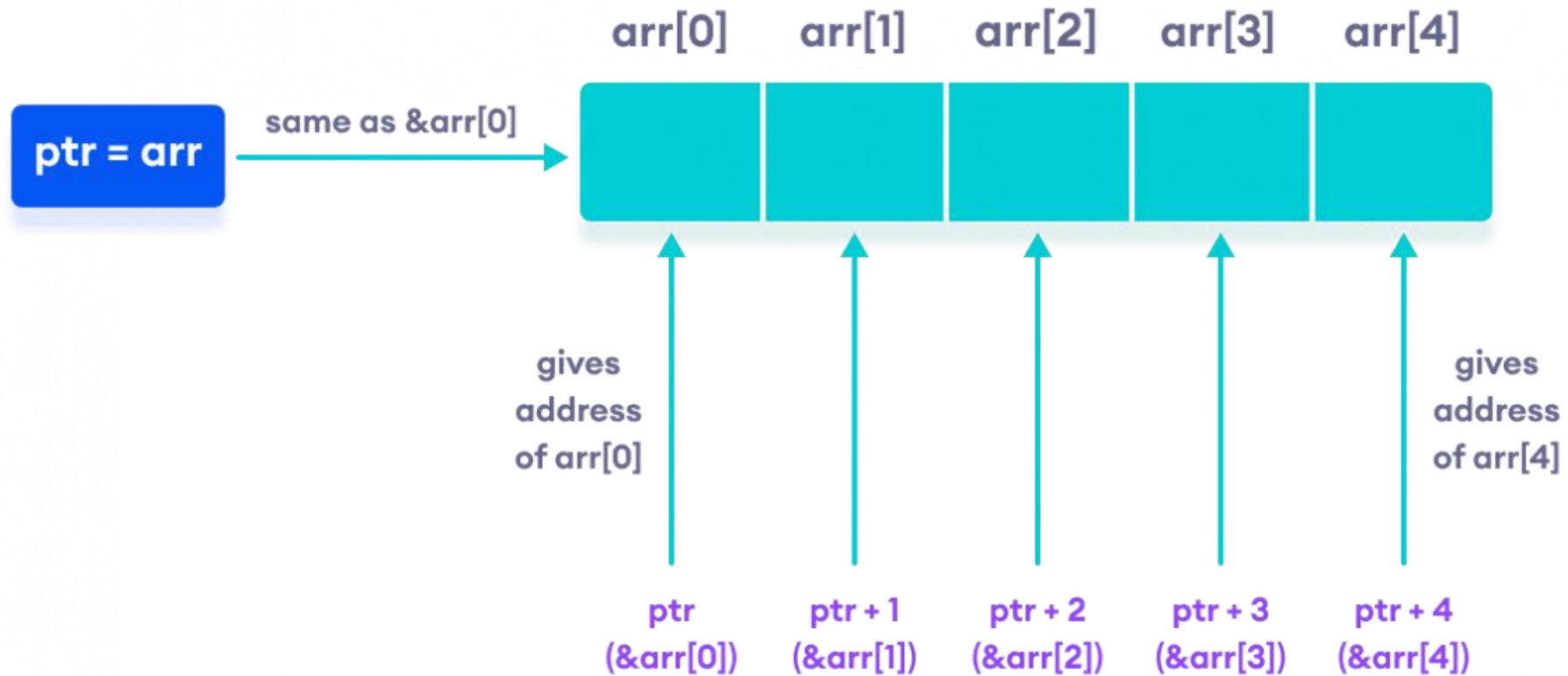
University of Utah, Salt Lake City, UT



Pointer



Array as Pointer!



Problem

Array Library

- Construct
- Destroy
- Read
- Print
- Add

Solution

```
1 #include <stdlib.h>
2
3 #include <iostream>
4
5 // Array
6 int* construct_array(int n) {
7     int* a = (int*)malloc(n * sizeof(int));
8     return a;
9 }
10
11 void destroy_array(int* a) { free(a); }
12
13 void read_array(int* a, int n) {
14     std::cout << "Enter " << n << " numbers: ";
15     for (int i = 0; i < n; ++i) std::cin >> a[i];
16 }
17
18 void print_array(int* a, int n) {
19     for (int i = 0; i < n; ++i) std::cout << a[i] << ' ';
20     std::cout << '\n';
21 }
22
23 int* add_array(int* a, int* b, int n) {
24     int* c = construct_array(n);
25     for (int i = 0; i < n; ++i) c[i] = a[i] + b[i];
26     return c;
27 }
```

Matrix as Array!

row, col

0,0	0,1
1,0	1,1
2,0	2,1

	0,0	0,1	1,0	1,1	2,0	2,1	
--	-----	-----	-----	-----	-----	-----	--

Matrix as Array!

$\text{ind} = \text{row} * \text{number-of-cols} + \text{col}$

0	1
2	3
4	5

	0	1	2	3	4	5	
--	---	---	---	---	---	---	--

Problem

Matrix Library

- Construct
- Destroy
- Read
- Print
- Add

Solution

```
30 // Matrix
31 int* construct_matrix(int m, int n) {
32     int* A = construct_array(m * n);
33     return A;
34 }
35
36 void destroy_matrix(int* A) { destroy_array(A); }
37
38 void read_matrix(int* A, int m, int n) { read_array(A, m * n); }
39
40 void print_matrix(int* A, int m, int n) {
41     for (int i = 0; i < m; ++i) print_array(A + i * n, n);
42 }
43
44 int* add_matrix(int* A, int* B, int m, int n) {
45     int* C = add_array(A, B, m * n);
46     return C;
47 }
```

Assignment

Matrix Multiplication

The diagram shows the multiplication of two matrices. The first matrix is $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ and the second matrix is $\begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}$. A yellow curved arrow labeled "Dot Product" connects the first row of the first matrix (1, 2, 3) to the first column of the second matrix (7, 9, 11). The result is shown as $= \begin{bmatrix} 58 & \end{bmatrix}$, with the value 58 highlighted in a yellow circle.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \end{bmatrix}$$

Solution

```
50 int* mul(int* A, int* B, int M, int N, int K) {
51     // Create matrix C[M][N]
52     ?
53     for (int m = 0; m < M; ++m) {
54         for (int n = 0; n < N; ++n) {
55             // Initialize C(m, n) to zero
56             ?
57             for (int k = 0; k < K; ++k) {
58                 // C(m, n) += A(m, k) * B(k, n)
59                 ?
60             }
61         }
62     }
63     return C;
64 }
```