CS 101 Computer Programming and Utilization

Lecture 17

2D arrays

Mar 15,16 Tuesday 2011

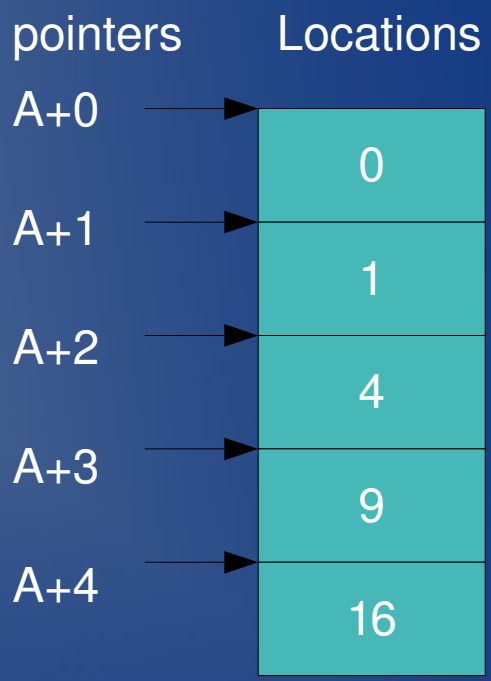
Prof. R K Joshi
Computer Science and Engineering
IIT Bombay
Email: rkj@cse.iitb.ac.in

Revision

- Type casting
  - converting types
  - type of result of a/b?
  - converting pointer types
- char pointers
- << and char pointers
- void pointers
- null pointers
- arguments to main
- converting the types of arguments to main
  - functions from cstdlib
- Introduction to string library
  - strings as objects
  - member functions on strings
  - resizable strings

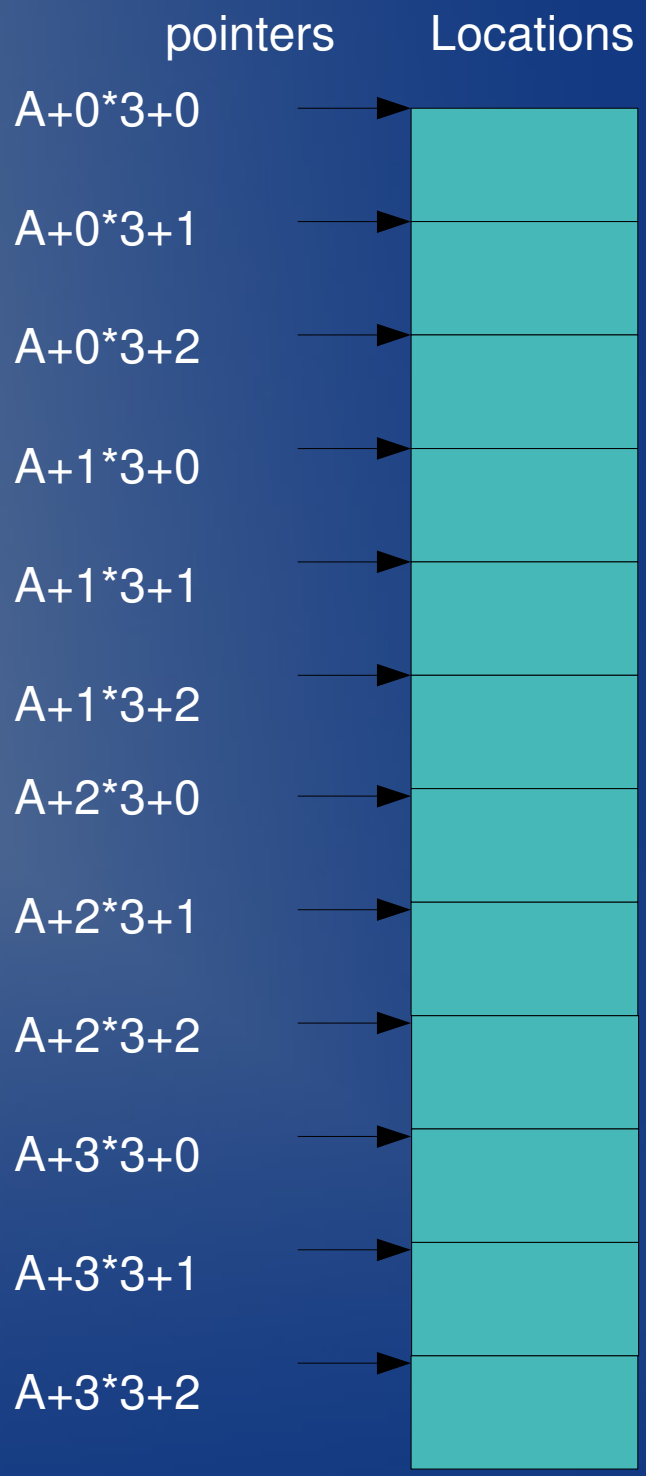
Pointer Arithmetic Again

```
int *A;
int n;
cin >> n;
A = new int[n];
for (i=0; i<n; i++)   {
        *(A+i)  = i*i;
}
```



2 Dimensional Array as 1D array

```
int *A;
int n;
cin >> m >> n; //4,3
A = new int[m*n];
for (i=0; i<m; i++)   {
   for (j=0; j<n; j++)  {
      *(A+i*n+j)  =random()%10;
  }
}
```



2D arrays with known dimensions

- int f (int P[2][4]) cannot accept A[2][3], but B[3][4] is okay
- 2nd dimension (no. of columns per row) is required to convert an access A[i][j] into a memory location:
  - in 1D: A[i] is same as *(A+i)
  - in 2D: A[i] [j] is same as *(A+i*no_of_columns+j)
- with 2nd dimension as 4 in one case and 3 in the other will give incorrect locations
- This problem does not arise in 1D arrays.