ESc101: Arrays

Instructor: Krithika Venkataramani

Semester 2, 2011-2012

The content of most slides are from the lecture slides of Prof. Arnab Bhattacharya and Prof. Dheeraj Sanghi

Arrays

- Array: collection of variables of the same type
- Convenient way to store and represent a list of variables
- An array is declared using [ ]
- Example: list of 100 integers
  - int a[100];
- Individual elements of the array are a[0], a[1], a[2], a[3], …, a[99]
  - Important: Array indexing starts with 0
  - If array a has n elements, there is no a[n]
  - Error with any index < 0 or ≥ n

Example program using arrays

- Read 10 numbers and printing them in reverse order.

```
int a[10], i;
printf ("Enter 10 numbers: ");
for (i = 0; i < 10; i++)
    scanf ("%d", &a[i]); // & is still there before a[i]
printf ("In reverse order: ");
for (i = 9; i >= 0; i--)
    printf ("%d ", a [i]);
```

Two-dimensional Arrays

- Arrays can be of arrays – this will become a 2-dimensional array.
- Declaration is by having the square brackets twice, [ ] [ ]
- Example: float x[5][6];
  - Name of the array is x.
  - Individual elements of array are x[0], x[1], x[2], x[3], and x[4], which themselves are arrays.
  - In this example, 30 (5 * 6) floating point numbers can be stored
  - These numbers can be accessed through x[i][j], for example, x[1][2].
  - First index, i, can vary from 0 to 4.
  - Second index, j, can vary from 0 to 5.
  - Convenient way to describe matrix

Array initialization

- The initial value of elements in an array can be given in the array declaration
- Example: int a[5] = {1, 2, 3, 4, 5};
  - a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4; a[4] = 5;
- 2D array Example: int a[2][3] = {1, 2, 3, 4, 5, 6};
  - int a[2][3] = {{1, 2, 3},{ 4, 5, 6}};
  - a[0][0] = 1; a[0][1] = 2; a[0][2] = 3; a[1][0] = 4; a[1][1] = 5; a[1][2] = 6;
- The least significant index in multi-dimensional arrays changes more frequently during initialization

Searching for a particular integer

- Search for a particular integer value in a sequence of 5 integers

```
int a[5], j, n;
for (j = 0; j<=4; j++)
{
    printf("Enter the number %d:",j+1);
    scanf("%d", &a[j]);
}
printf("Enter the number to be searched: ");
scanf("%d", &n);
```


Searching for a particular integer (cont.)

```
for (j =0; j<=4; j++)
{
    if (a[j] ==n)
    {
        printf("Number found at index %d", j+1);
        //no break to find the number, if present, multiple times
    }
}
```

Transpose of a matrix

```
double matrix [4][4];
int i, j;
for (i = 0; i < 4; i++)
    for (j = 0; j < 4; j++)
        scanf ("%f", &matrix[i][j]); // Read one element of matrix
printf ("The transpose matrix is:\n");
for (i = 0; i < 4; i++)
{
    for (j = 0; j < 4; j++)
        printf ("%f\t", matrix[j][i]); // Write indices in reverse order
printf ("\n"); // Works only for square matrix
}
```


Transpose of a non-square matrix

```
double matrix [3][4];
int i, j;
for (i = 0; i < 3; i++)
    for (j = 0; j < 4; j++)
        scanf ("%f", &matrix[i][j]); // Read one element of matrix
printf ("The transpose matrix is:\n");
for (j = 0; j < 4; j++) // Interchanging the loops works
{
    for (i = 0; i < 3; i++)
        printf ("%f\t", matrix[i][j]);
        // Indices should not be interchanged
printf ("\n");
}
```


Multi-dimensional arrays

- Arrays of 3 or more dimensions are allowed
  - Example: int array[2][4][6] – total numbers = 2*4*6 = 48
- Limit on number of dimensions is compiler dependent
- Can be character arrays too
- The array dimensions need to be specified
- Example: Find mean and standard deviation of All (3) quiz marks of 12 sections having 30 students per section

Sample program with multi-dimensional arrays

- Find section-wise average of 3 quiz marks in a class having 12 sections. Each section has 30 students each. The input is in a file, formatted as follows.
  - Sec 1, Student1: q1marks q2marks q3marks
  - ...
  - Sec 1, Student30: q1marks q2marks q3marks
  - Sec 2, Student1: q1marks q2marks q3marks
  - ...
  - Sec 2, Student30: q1marks q2marks q3marks
  - ...
    - 10 5 3
    - ...
    - 8 5 9

Section-wise average for each of 3 quizzes

```
#include<stdio.h>
void main()
{
    float marks[12][30][3], average[12][3];
    int i, j, k;
    //read all marks from input
    for(i=0; i<12; i++)//Reading section-wise
        for(j=0;j<30; j++)//reading student-wise
            for(k=0;k<3;k++)//reading quizzes of a student
                scanf("%f",&marks[i][j][k]);
```



```
    for(i=0; i<12; i++) //averaging section wise
    {   printf("Average of Section %d - ",i+1);
        for(k =0; k<3; k++) //averaging for each quiz
        {
            average[i][k]= 0.0; //initializing the quiz-average
            for(j=0;j<30;j++)
            {
//updating quiz(k+1)th average with marks of all students in section(i+1) in this loop
                average[i][k]= average[i][k]+marks[i][j][k];
            }
            average[i][k] = average[i][k]/30;
            printf("Quiz%d: %f\t",k+1,average[i][k]); /*averages of each quiz*/
        }
        printf("\n"); // new line for each section
    }
} // end of main()
```