

# CSE 142 Programming I

## Multidimensional Arrays

Textbook 8.7

© 2000 UW CSE

5/12/00

M-1

## Arrays as Data Structures

- Review: An array is an ordered collection of values of identical type
  - Name the collection; number the elements
- Arrays are the natural choice for organizing a large number of values, all of identical type
- Sometimes the values already have some additional regular pattern or structure
- One common such structure is the "matrix" or two-dimensional array

5/12/00

M-2

## 2-Dimensional Arrays

- An ordered collection of values of identical type
- Name the collection; number the elements
  - Like 1-D arrays, but a more complicated numbering scheme
- Example: scores for 7 students on 4 homeworks

score	hw	0	1	2	3
student 0		22	15	25	25
student 1		12	12	25	20
student 2		5	17	25	24
student 3		15	19	25	13
student 4		2	0	25	25
student 5		25	22	24	21
student 6		8	4	25	12

C expressions:

`score[0][0]` is 22  
`score[6][3]` is 12  
`2*score[3][0]` is 30

5/12/00

M-3

## Declaring a 2D Array

```
#define MAX_STUDENTS 80
#define MAX_HWS 6
...
int score [MAX_STUDENTS] [MAX_HWS] ;
int nstudents, nhws, i, j ;
```

5/12/00

M-4

## 2D Arrays: Two Views

```
type name[#rows][#columns]
```

```
int score[80][6]
```

score is a two-dimensional array of int of size 80 by 6

or... score is an array of size 80 of type array of size 6 of int

score[0][0], score[0][1], ..., score[79][5] are the elements of the array

or... score[0], score[1], ..., score[79] are the rows of the array and each of the rows is itself an array.

5/12/00

M-5

## Reading a 2D Array

```
scanf ("%d %d", &nstudents, &nhws) ;
if (nstudents <= MAX_STUDENTS &&
    nhws <= MAX_HWS) {
    for ( i = 0 ; i < nstudents ; i = i + 1 )
        for ( j = 0 ; j < nhws ; j = j + 1 )
            scanf ("%d", &score [i] [j]) ;
}
```

Part of the array is unused; which part?

5/12/00

M-6

## Array Input Example

Input: 7 4 0 1 2 3 4 5 6 7 8 9 ...

```

score  j= 0 1 2 3 4 5 ...
i=0    0 1 2 3 ? ? ...
i=1    4 5 6 7 ? ? ...
i=2    8 9 ...
...
i=6    ...
i=7    ? ? ? ? ...
    
```

5/12/00

M-7

## Printing a 2D Array

```

if (nstudents <= MAX_STUDENTS &&
    nhws <= MAX_HWS) {
    for (i = 0; i < nstudents; i = i + 1) {
        for (j = 0; j < nhws; j = j + 1)
            printf("%d", score[i][j]);
        printf("\n");
    }
}
    
```

5/12/00

M-8

## 2D Arrays as Parameters

```

void read_2D (int a [MAX_STUDENTS] [MAX_HWS],
             int nstudents, int nhws)
{
    int i, j;
    for (i = 0; i < nstudents; i = i + 1)
        for (j = 0; j < nhws; j = j + 1)
            scanf("%d", &a[i][j]);
}
    
```

5/12/00

M-9

## Array Function Arguments

```

int main(void)
{
    int score [MAX_STUDENTS] [MAX_HWS];
    int nstudents, nhws;

    scanf ("%d %d", &nstudents, &nhws);
    if (nstudents <= MAX_STUDENTS &&
        nhws <= MAX_HWS)
        read_2D (score, nstudents, nhws);
    ...
}
    
```

no &

5/12/00

M-10

## 2-D Score Averages

```

int score [MAX_STUDENTS] [MAX_HWS], total [MAX_HWS];
int nstudents, nhws, i, j;
/* Input nstudents and nhws. Use read_2D to input scores. [code omitted] */
/* Calculate and print averages for each homework assignment */
for (j = 0; j < nhws; j = j + 1) {
    total[j] = 0;
    for (i = 0; i < nstudents; i = i + 1)
        total[j] = total[j] + score[i][j];
}
printf("Averages: \n");
for (j = 0; j < nhws; j = j + 1)
    printf("HW %d: %f \n", j, (double) total[j] / (double) nstudents);
    
```

5/12/00

M-11

## 2D Array Elements as Parameters

/\* Interchange rows row1 and row2 of array a \*/

```

void swap_rows (int a [][MAX_HWS],
               int row1, int row2, int ncols)
    
```

```

{
    int j;
    for (j = 0; j < ncols; j = j + 1)
        swap(&a[row1][j], &a[row2][j]);
}
    
```

	0	1	2	3	4	5
0	13	15	25	25	?	?
1	12	12	25	20	?	?
2	5	17	25	24	?	?
3	15	19	25	13	?	?
4	2	0	25	25	?	?
5	25	22	24	21	?	?
6	8	4	25	12	?	?
7	?	?	?	?	?	?

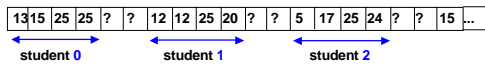
5/12/00

M-12

## Representation of Arrays

(Or: can you omit dimensions in a formal parameter?)

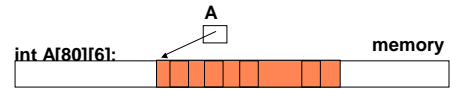
score	hw	0	1	2	3	4	5
student 0		13	15	25	25	?	?
student 1		12	12	25	20	?	?
student 2		5	17	25	24	?	?
student 3		15	19	25	13	?	?
student 4		2	0	25	25	?	?
student 5		25	22	24	21	?	?
student 6		8	4	25	12	?	?
student 7		?	?	?	?	?	?



5/12/00

M-13

## A 2D Array as a Pointer



$A[i]$  equivalent to  $*(A + 6 * i)$

$A[i][j]$  equivalent to  $*(A + 6 * i + j)$

5/12/00

M-14