

"Thursday" Extras this week:

Today, October 30 @ noon

Pizza, drinks, and treats from Hungary provided

Prof. David Szeszler

AIT Budapest Program

Thursday, October 31 @ 4:00 pm

Computer Science at Grinnell (Advising, CS Major, etc.)

Clarification: Option Test Revision due at start of class (8:30 am), Wednesday, November 6

Do not even think about turning it in at 8:35 am or later on next Wednesday!!!!

Times based on MathLAN computers!!!

Reminder: When working in pairs, switch often

(e.g., every step, every section, at least every day)

Pre-registration suggestions

CSC 207: completes sequence of multiple views of prob. sol.

CSC 213: operating systems—builds on CSC 161

CSC/MAT 208: Discrete Structures— math essential for later CS

(if not taking MAT 218, Discrete Bridges to Adv Mathematics)

CSC 261: Artificial Intelligence

(Requires both CSC 161 and either CSC 208 or MAT 218)

CS Major notes

Talk to any CS faculty member (e.g., me), if you have questions about the CS major (e.g., courses, schedule, options, study abroad, etc.)

Since staffing still evolving for CS,

second-year students cannot declare CS major until spring semester

2-dimensional arrays

questions

clicker questions

What is printed by this code?

```
char table[3][4] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };  
  
printf("%c",table[1][2]);
```

A. b

B. e

C. g

D. j

E. k

```
char table[3][4] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };
```

Address	Value
table	a
table+1	b
table+2	c
table+3	d
table+4	e
table+5	f
table+6	g
table+7	h
table+8	i
table+9	j
table+10	k
table+11	l
table+12	???
table+13	???
table+14	???
15	???

Which of the following declarations are valid

A. 1, 2, and 3

B. only 1

C. only 2

D. only 3

E. some combination of two

```
1. char table[ ][4] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };
```

```
2. char table[3][ ] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };
```

```
3. char table[ ][ ] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };
```

Which of the following declarations are valid

- A. 1, 2, and 3
- B. only 1
- C. only 2
- D. 1 and 3
- E. 2 + one other

1. `char table[ ][7] =`
 `{ {'a', 'b', 'c', 'd'},`
 `{'e', 'f', 'g', 'h'},`
 `{'i', 'j', 'k', 'l'} };`

2. `char table[4][2] =`
 `{ {'a', 'b', 'c', 'd'},`
 `{'e', 'f', 'g', 'h'},`
 `{'i', 'j', 'k', 'l'} };`

3. `char table[4][7] =`
 `{ {'a', 'b', 'c', 'd'},`
 `{'e', 'f', 'g', 'h'},`
 `{'i', 'j', 'k', 'l'} };`

What is printed by this code?

```
char table[3][4] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };  
  
printf("%c", *(table+6));
```

- A. f
- B. j
- C. g
- D. Compiler error
- E. Segmentation fault

What is printed by this code?

```
char table[3][4] =  
    { {'a', 'b', 'c', 'd'},  
      {'e', 'f', 'g', 'h'},  
      {'i', 'j', 'k', 'l'} };
```

```
printf("%c",table[1][6]);
```

1. b
2. e
3. g
4. j
5. k
6. Compiler error
7. Run-time error

Main points regarding 2-dimensional arrays

- Arrays are fixed size
 - Both # rows and #columns specified
 - Technically, a 2D array is a 1D array of 1D arrays
- 2D arrays only store elements of the same type
- Arrays are stored row by row

Given declaration `some_type array[numRows][numCols]`

- All rows have the same width, making access very fast!
- Access (e.g., `array[i][j]`) is through base address + offset
 - $\text{offset} = (\text{sizeofRow} * i) + j$ (based on size of `some_type`)
 - `array` is base address of the overall array
 - `array[i]` is the base address of row `i`
 - `array[i][j]` is a specific element within the array
 - C does not perform bounds checking, so no access overhead