Today's plan

- Class interactions
    - Ex 3-2
    - Keys points
    - Recap discussion
- Class exercises
    - Ex 3-3

Ex 3-2

Volunteers?

- C-string manipulation - “null terminator”.
- Separate complication and header guards.
- Write a Makefile.
    - targets
    - variables
    - commands

Key points - multi-dimensional arrays

-
  ```
  int foo[20][50];
  ```

  ```
  int bar[3][10] = {
        {1,2,3,[5]=11,12,13}
      , [2]={10,9,8,[5]=-1,-2,-3}
  };
  ```

- What is printed?

  ```
  for (int y = 0; y < 3; ++y) {
      for (int x = 0; x < 10; ++x)
          printf("%d ", bar[y][x]);
      printf("\n");
  }
  ```

  ```
  1 2 3 0 0 11 12 13 0 0
  0 0 0 0 0 0 0 0 0 0
  10 9 8 0 0 -1 -2 -3 0 0
  ```

Key points - multi-dimensional arrays

-
  ```
  int foo[20][50];
  int bar[3][10] = {
      {1,2,3,[5]=11,12,13}
      , [2]={10,9,8,[5]=-1,-2,-3}
  };
  ```
- What is printed?
  ```
  for (int y = 0; y < 3; ++y) {
      for (int x = 0; x < 10; ++x)
          printf("%d ", (int*)(bar)[y * 3 + x]);
      printf("\n");
  }
  ```
  ```
  1 2 3 0 0 11 12 13 0 0
  0 0 0 0 0 0 0 0 0 0
  10 9 8 0 0 -1 -2 -3 0 0
  ```
- Come back to this after we learn pointers next week!

GDB

- Compiler flag '-g': enable debugging symbols (for tracing the program using GDB).
- Run GDB to debug a program: `gdb <executable>`.
- Run GDB to debug a program with cmd-line arguments: `gdb --args <executable> <args ...>`.
- Common steps using GDB:
  - 'break' to set a break point if you know where to stop. e.g. `break <filename>:<line number>`.
  - 'run' to execute the program. It will stop at your break point or where the program crashes.
  - 'list' to show the current code block.
  - 'backtrace' to show the call stacks, which shows where it crashes and how it reached there.
  - 'print' to print the value of a variable or 'display' to monitor the value per each step.

GDB - cont

- Common steps using GDB cont:
  - ‘next’ to step through the program and check the displayed variable values.
  - ‘step’ to go into a function if needed.
  - ‘finish’ to step out the current function.
  - ‘continue’ to run the program. It will stop at the next break point or where it crashes if any.
  - ‘record’ to start recording your instructions for reverse if needed.
  - ‘reverse-next’ to reverse the previous next.
  - ‘reverse-step’ to reverse the previous step.
  - ‘reverse-finish’ to reverse the previous finish.
  - ‘reverse-continue’ to reverse the previous continue.
  - ‘quit’ to exit GDB.

Check the gdb cheat sheet and find the command to print the content of a variable per step, instead of only printing it once using `print'.

Mentimeter

p var

display

display var

4

Hod do you declare a multi-dimensional array and pass it to a function?

Mentimeter

function(<type>[][]) ✕

<base_type> <name> [dim1_z] [dim2_z]. When passing to a function, you have to specify at least the second and following dimensions ✕

<type> <name>[][] ✕

type multi_array[dim_1][dim_2] ✕

The correct answer is: e.g. int foo[2][3]; void bar(int foobar[][3]); bar(foo);

4

Mentimeter

How do you initialize a multi-dimensional array using array initialization?

array[][]

type[][] = {[0,0,0],[0,0,0],[0,0,0]}

Int arr[2][3] = {{1, 2, 3}, {5, 6, 7}};

eg. int table[2][4] = { {1,2,3,4},{5,6,7,8} };

The correct answer is: e.g. int foo[2][3] = { {1, 2, 3}, {4, 5, 6} };

4

What is the compile flag needed to compile a program such that we can debug it using GDB?

Mentimeter

-g ✓ 6x

The correct answer is: -g

6

How do you set a break point using GDB and check the call stack?

Mentimeter

break point: break <where>, check call stack: where

break <where> ; backtrace

break <filename>:<linenumber>

Set a break point with "break" and check the call stack with "backtrace"

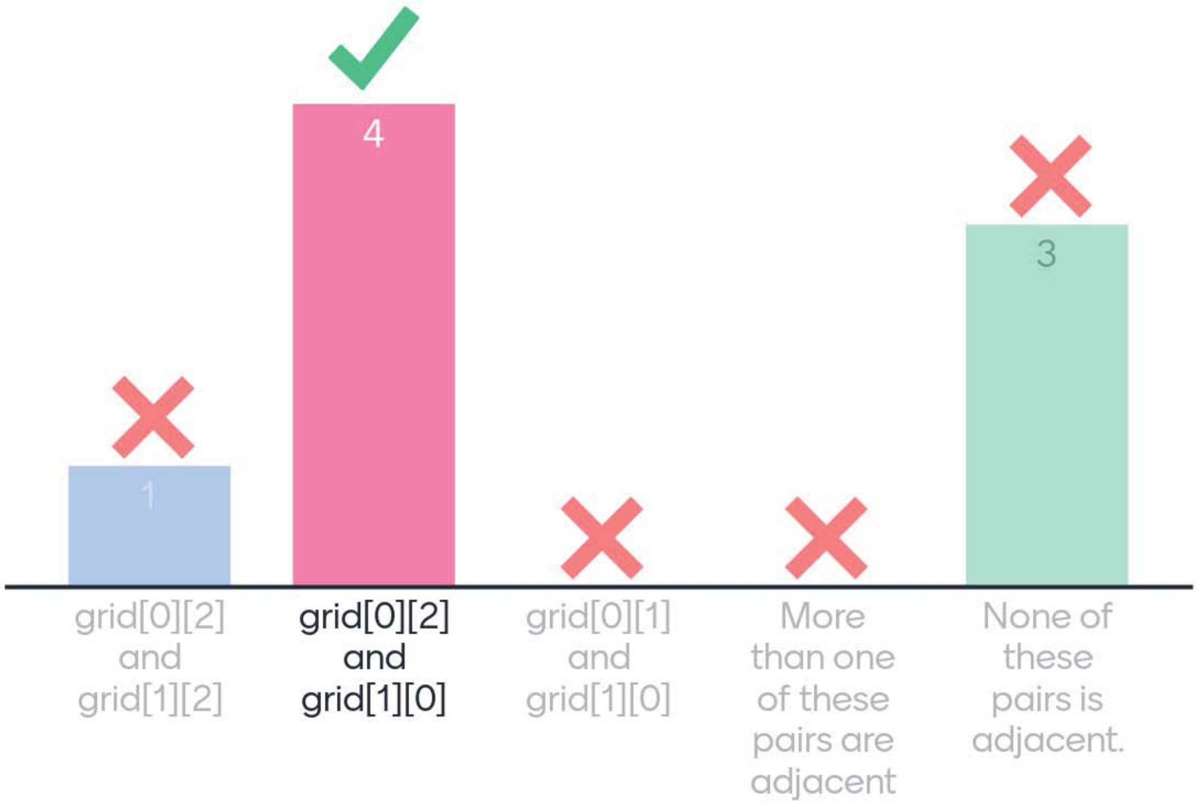
break <file name><line number>

The correct answer is: break <filename>:<line number>, then backtrace

5

Consider this array declaration: `float grid[2][3];'. Which of the following pairs of elements are adjacent in memory?

Mentimeter



8

Class exercises

Ex 3-3