Slides + code available at
www.github.com/pkorth/eecs280_recitation

EECS 280

Discussion 04: Feb 04, 2015

MichiganEngineering

Slides + code available at
www.github.com/pkorth/eecs280_recitation

Agenda

- Logistics

- Brief review of lecture material
  - Arrays
  - Array traversal using indexing
  - Pointers
  - Array traversal using pointers
  - Not covered: const, C-strings (next week)

- Work on Lab 04

MichiganEngineering

Logistics

- Lab 04
  - Array traversal
  - No pre- or post-lab survey
  - Due Friday 2/6

- Project 3
  - Euchre simulator (a Midwestern card game)
  - Pointers, arrays, structs, polymorphism
  - Checkpoint due in two weeks, Thursday 2/19

Slides + code available at
www.github.com/pkorth/eecs280_recitation

Logistics

- Editor/IDE Guide is being published this week
- Midterm in three weeks, Wednesday 2/25

Arrays

A container to hold multiple instances of a data type

- Set of Players in a card game
- The stock price of GOOG for the past 100 days
- Names of all employees at Mujo Cafe

```
double goog_stock[100];
goog_stock[0] = 529.24;
goog_stock[1] = 528.48;
// ...
```

Arrays

We can pass them to functions

```
void print_row(char row[], int size) {...}
```

We can make them multidimensional

```
char checker_board[8][8];
checker_board[0][0] = ‘B’;
checker_board[4][2] = ‘R’;
```

Arrays: 1D

checkers[column]

Arrays: 1D traversal with indexing

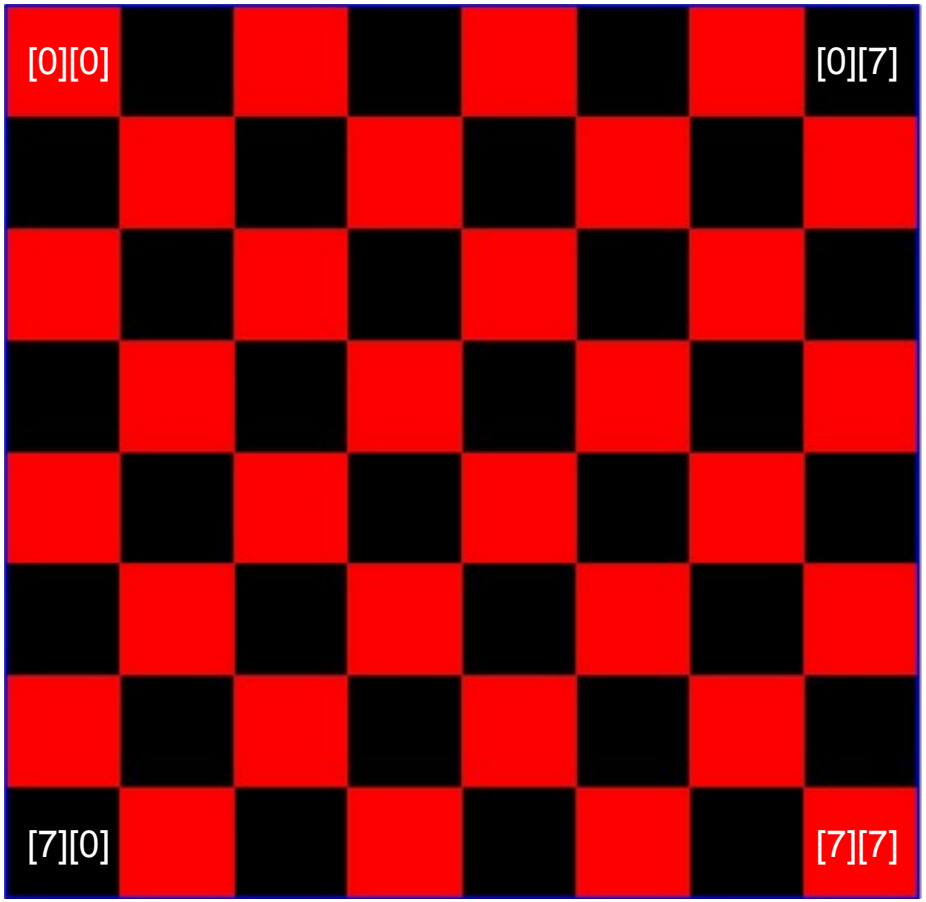
```
char checkers[8];

for(int i = 0; i != 8; i++) {
  checkers[i] = ' ';
}
```

Our container

Which column? (x)

Arrays: 2D

checkerboard[row][column]



Arrays: 2D traversal with indexing

```
char checker_board[8][8];

for(int i = 0; i != 8; i++) {
  for(int j = 0; j != 8; j++) {
    checkerboard[i][j] = ' ';
  }
}
```

Our container

Which row? (y)

Which column? (x)

Pointers: conceptual

Variables we've worked with so far hold a value.

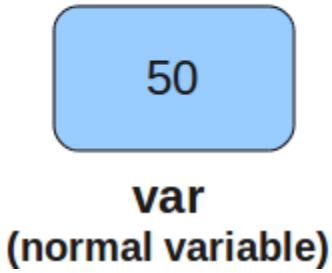


Image:
www.thegeekstuff.

Pointers: conceptual

The computer has to store them in memory somewhere.

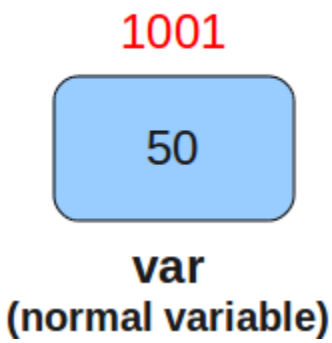


Image:
www.thegeekstuff.

Pointers: conceptual

A pointer is a variable that holds the memory address of another variable. It “points” to another variable.

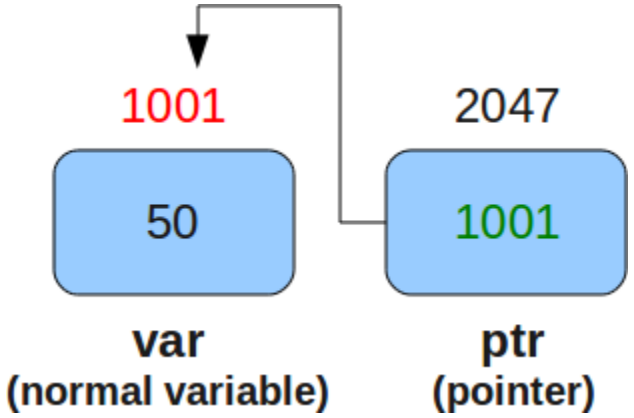


Image:
www.thegeekstuff.

Pointers

```
<pointer type> *<pointer-name>;
<pointer-name> = &<variable-name>;
*<pointer-name> = value;
```

```
int temperature = 27;
int *ptr = &temperature;
*ptr = 24;
```

Pointers

We can pass them to functions

```
void magic(int *ptr) {...}
```

Pointers can point to a spot in an array

```
double values[8];
double *ptr;
ptr = values;
ptr = &(values[5]);
```

Array traversal: pointers

```
char checkers[8];

for(char *ptr = checkers; ptr != checkers+8;
                                    ptr++) {
  *ptr = ' ';
}
```

Lab 04

Goal

Practice writing array traversal using indexing and pointers

Tasks

1. Implement printRowIndex() and printRowPtr()
2. Implement three more array functions (optional)