CS50 Section. Week 4. 10/6/15.

Tuesdays 7:00-8:30pm, Science Center 309A

Neel Mehta
neelmehta@college.harvard.edu
(215) 990-6434

Get these handouts at https://github.com/hathix/cs50-section/tree/master/handouts.

Quiz 0

Check out the handout from our Quiz 0 Review: https://github.com/hathix/cs50-section/blob/master/handouts/quiz0-review.md.

Recursion

pow (Fall 2011 Quiz 0)

Complete the implementation of `pow` below in such a way that the function returns x^y (i.e., `x` raised to the power of `y`) unless `x` or `y` (or both) is negative, in which case the function should instead return `-1`. Recall that, mathematically, x^0 is 1 and that x^1 is x. You needn't worry about integer overflow. Suffice it to say that you may not call the version of `pow` that's declared in `math.h`!

```
int pow(int x, int y)
{









}
```

Solution: http://cdn.cs50.net/2011/fall/quizzes/0/key0.pdf

Fibonacci

We can define the Fibonacci numbers as such:

- The 0th and 1st Fibonacci numbers are both 1.

- Every subsequent Fibonacci number is the sum of the previous 2 Fibonacci numbers.

So the first few Fibonacci numbers are:

```
Index   0  1  2  3  4  5
Number  1  1  2  3  5  8
```

Write a recursive function that calculates the nth Fibonacci number. Assume you'll only be given positive numbers. **You can't use loops.**

- `fibonacci(5)` should return `8`
- `fibonacci(1)` should return `1`
- `fibonacci(0)` should return `1`

```
int fibonacci(int index)
{









}
```

Solution: https://github.com/hathix/cs50-section/blob/master/code/5/fibonacci-soln.c

Sums

Write a recursive function `sum` that, given an array `numbers` of length `length`, finds the sum of the numbers starting at index `start_index`. **You can't use loops.**

If `int x[] = {5, 6, 7}` and `int y[] = {}`, then:

- `sum(x, 3, 0)` should return `18` (5 + 6 + 7)
- `sum(x, 3, 1)` should return `13` (6 + 7)
- `sum(y, 0, 0)` should return `0`

```
int sum(int numbers[], int length, int start_index)
{









}
```

Solution: https://github.com/hathix/cs50-section/blob/master/code/5/sums-soln.c

Pointers

Quick review

There are three fundamental pointer operations, illustrated here with `a`, `b`, and `c`. Explain what each operation does.

Dereference

```
//
//
int x = *a;
```

Address of

```
//
//
int* x = &b;
```

Assignment

```
//
//
*c = 5;
```

swap

Finish this code that'll swap the values of two integers.

```
void swap(int* a, int* b)
{
    // YOUR CODE HERE









    // END YOUR CODE
}

int main(void)
{
    int x = 1;
    int y = 2;

    // prints "x = 1, y = 2"
    printf("x = %i, y = %i\n", x, y);

    // call your swap function
    // YOUR CODE HERE


    // END YOUR CODE

    // prints "x = 2, y = 1"
    printf("x = %i, y = %i\n", x, y);
}
```

Solution: https://github.com/hathix/cs50-section/blob/master/code/5/swap-soln.c

strlen (Fall 2013 Quiz 0)

Suppose that you've forgotten which header file declares `strlen`, and so you need to re-implement it yourself. Bah. Even worse, neither `[` nor `]` currently works on your keyboard (or pencil or pen). Without calling any functions at all and without using any square brackets, complete the implementation of `strlen` below **using pointer arithmetic** in such a way that the function returns the length of `s`. If `s` happens to be `NULL`, your implementation of `strlen` should return 0.

```
int strlen(char* s)
{









}
```

Solution: http://cdn.cs50.net/2013/fall/quizzes/0/key0.pdf

File I/O

Opening and closing files

https://reference.cs50.net/stdio.h/fopen, https://reference.cs50.net/stdio.h/fclose

```
FILE* file_pointer = fopen(filename, "r");
// do stuff
fclose(file_pointer);
```

Reading from files

https://reference.cs50.net/stdio.h/fread

Reading into an array:

```
int length = 50;
int destination[length];
fread(&destination, sizeof(int), length, file_pointer);
```

Reading into a single variable:

```
int destination;
fread(&destination, sizeof(int), 1, file_pointer);
```

Writing to files

https://reference.cs50.net/stdio.h/fwrite

Writing from an array:

```
int length = 50;
int source[length];
fwrite(&source, sizeof(int), length, file_pointer);
```

Writing from a single variable:

```
int source;
fwrite(&source, sizeof(int), 1, file_pointer);
```

Moving file pointer

https://reference.cs50.net/stdio.h/fseek

```
int distance = 50;
fseek(file_pointer, distance, SEEK_CUR);
```