C Language Concepts

CS 231

functions

Math Library Functions

- *#include <math.h>*

- *functions include sqrt, exp, log, ceil, floor, cos*

Pass by Value or Pass by Reference

* *C always uses pass by value*

    * *A copy of the argument is made, and the copy is passed to function*

* *Some languages allow pass by reference*

    * *A change to parameter also changes the argument*

* *Simulate pass by reference in C*

    * *Create a pointer to variable and pass pointer to function*

    * *If function changes variable pointed to, this simulates passing variable by reference*

Recursion

- *A function which calls itself is recursive*

- *This works because each time a function is called all local variable storage is placed on stack in a stack frame (which also contains control information)*

- *facts.h computes factorials*

  - *Discussion using idea of stack frames*

Recursive Solution to Finding nth Fibonacci Number

- *fib(0) = 0*

- *fib(1) = 1*

- *for n > 1, fib(n) = fib(n-1) + fib(n-2)*

- *fibo.c has code*

- *Discussion*

Problem to Solve

- *Consider the factorial function, n!*

- *Write a function to print the number of ending zeroes of n!*

  - *5! = 120, result should be 1*

  - *10! ends with 2 zeroes, answer is 2*

  - *20! ends with 4 zeroes (our fact.c code ends before this)*

  - *25! ends with 6 zeroes*

Solution Idea

- *Each time we multiply by a multiple of 5 we get an additional ending 0*

- *Multiplying by multiples of 25 gives two ending zeroes*

- *Multiplying by multiples of 125 gives 3 ending zeroes, etc.*

- *recursively divide n by 5 and add the integer results of division*

Functions as Arguments

* Problem: send out e-mail to all people on list matching certain criteria (criteria will change, process of sending a message in an e-mail does not).

  * To anyone one who is over 18 (need age)

  * To anyone who is over 21 (still need age)

  * To anyone who is between 16 and 40 (range of ages)

  * To men between 18 and 24 (range, but only men)

  * To women between 16 and 30

  * To any married person with income of at least $50,000

Difficulties With Solving Problem

- *If we have a function to send e-mail and want to pass criteria, many parameters are required, but not all used in any one call*

- *Function will be complicated in determining how to use criteria to select person to send e-mail to*

Higher-Order Functions in C

- *A way to solve this problem is to pass in a function which will select all people from list who meet all the requirements for this batch of e-mail*

- *Functions which take functions as arguments, or which return a function are higher order*

Declaring a function pointer

- *Consider function declaration below:*

  - *int f(int, double)*

- *Now declare a pointer to such a function:*

  - *int (*f)(int, double)*

- *functionsAsArg.c shows an example of this use.*