



Lab 11: Pointers

CSE 4108

Structured Programming I Lab

November 2022



Lab Tasks

Recursion:

1. Selection Sort:

Write a program that asks the user to enter a series of integers (which it stores in an array), then sorts the integers by calling the function **selection_sort**. When given an array with n elements, **selection_sort** must do the following:

(a) Search the array to find the largest element, then move it to the last position in the array.

(b) Call itself recursively to sort the first $n - 1$ elements of the array.

2. With Great Power, Comes Great Responsibility!:

power.c (Can be found on Chapter 9, page 205) can be made faster by having it calculate x^n in a different way. We first notice that if n is a power of 2, then x^2 can be computed by squaring. For example, x^4 is the square of x^2 , so x^4 can be computed using only two multiplications instead of three. As it happens, this technique can be used even when n is not a power of 2. If n is even, we use the formula $x^n = (x^{n/2})^2$. If n is odd, then $x^n = x \times x^{n-1}$. Write a recursive function that computes x^n . (The recursion ends when $n = 0$, in which case the function returns 1).

To test your function, write a program that asks the user to enter values for x and n , calls **power** to compute x^n , and then displays the value returned by the function.



3. GCD:

Write a program in C to find GCD of two numbers using recursion.

4. LCM:


Write a program in C to find LCM of two numbers using recursion.

Pointers:

5. Pointers:

Write a program that declares and initializes (to any value you like) a double, an int, and a char. Next declare and initialize a pointer to each of the three variables. Your program should then print the address of, and value stored in, and the memory size (in bytes) of each of the six variables. Here's an example of the desired output:

```
The address of char c is 0x61feff
The address of int i is 0x61fef4
The address of double d is 0x61fee8
The address of char* pc is 0x61fef8
The address of int* pi is 0x61fef0
The address of double* pd is 0x61fee4
The value of char c is a
The value of int i is 1
The value of double d is 2.500000
The value of char* pc is 6422271
The value of int* pi is 6422260
The value of double* pd is 6422248
The size of char is 1
The size of int is 4
The size of double is 8
The size of char* is 4
The size of int* is 4
The size of double* is 4
```



Note: Use the “0x%x” format specifier to print the addresses in hexadecimal.

Hint: Use the function `sizeof()` to get the memory size.

6. Swap:

Write a program that contains the following function:

```
void swap (int *x, int *y);
```

The function will swap the values pointed by x and y. The main function will pass addresses of two variables which need to be swapped.

7. Coin Change:

In **Lab 1 - C Fundamentals** (*Problem: Counting Denominations*), you wrote a program that takes the U.S. dollar amount as input and then shows how to pay that amount using the smallest number of \$20, \$10, \$5, and \$1 bills. Modify the program so that it includes the following function:

```
void pay_amount (int dollars , int *twenties , int *tens ,  
                int *fives , int *ones);
```

The function determines the smallest number of \$20, \$10, \$5, and \$1 bills necessary to pay the amount represented by the dollars parameter. The twenties parameter points to a variable in which the function will store the number of \$20 bills required. The tens, fives, and the parameters are similar.



8. Welcome Aboard - Pointers:

In **Lab 4 - Selection Statements** (*Problem: Welcome Aboard*), you wrote a program that asks users to enter a time (expressed in hours and minutes, using the 24-hour clock). Modify the program so that it includes the following function:

```
void find_closest_flight (int desired_time , int *departure_time  
                        , int *arrival_time);
```

The function will find the flight whose departure time is closest to `desired_time` (expressed in minutes since midnight). It will store the departure and arrival times of this flight (also expressed in minutes since midnight) in the variables pointed to by `departure_time` and `arrival_time`, respectively.

9. Reduce Fractions Saga - New Pointers:

In **Lab 5 - Loops** (*Problem: Reduce the Fractions*), you wrote a program that asks the user to enter a fraction, then reduces the fraction to lowest terms. Modify the program so that it includes the following function:

```
void reduce (int numerator , int denominator , int *  
            reduced_numerator , int * reduced_denominator);
```

`numerator` and `denominator` are the numerator and denominator of a fraction. `reduced_numerator` and `reduced_denominator` are pointers to variables in which the function will store the numerator and denominator of the fraction once it has been reduced to lowest terms.