Kent Mark

Cpre 288 – Homework 2

2/16/20

Homework 2

1. The question that I chose to answer was question 4 which focused on C functions, pointers, and using loops to iterate through a data structure and summing the individual elements. This question required me to complete a function that would take an integer array of numbers and tally the sum of its members. Solution with in-line comments follows as such:

#include “stdio.h”

void get\_sum(int \*sum, int num\_array[], int size){

int j; // variable to serve as iterant to iterate through array

\*sum = 0; // method returns void; sum will be stored as pointer reference

for(j = 0; j < size; j++){ // j iterates from index 0 to final index

\*sum = \*sum + num\_array[j]; // every element j is added to current sum, then stored as a pointer reference

}

}

main(){

int array\_sum; // int variable to store sum of array

int my\_array[] = {25, 18, 6, 47, 2, 73, 100}; // array that will be fed into function

get\_sum(&array\_sum, my\_array, 7\_); // method takes the address of the empty variable array\_sum, the actual int array, the array’s bounds for the for loop

printf(“sum = %d”, array\_sum); // statement returns the sum the int array

//Desired output of printf: sum = 271

}

1. A key concept from class that can be related to this question would be “Pointers”. In class we learned that pointers are special variable that can be used to store the memory address of another variable. We also learned that using the “&” operator broadcasts the address of the variable in system memory that can then be both accessed and modified by a pointer variable.
2. A resource that I used to help me answer the homework questions was the “C for Dummies: All in One Desk Reference” written by Dan Gookin. It provided me with some great explanations of the meaning and uses of pointers, the “\*” operator to declare a pointer variable, and the “&” that allows said pointer to copy another variable’s memory address and contents.
3. My own question:

Main(){

int num = 100;

int \*ptr;

ptr = &num;

printf(“value of num is %d\n”, \*ptr);

return 0;

}

Output: value of num is 100