***Project 6 - Due November 27***

1.

a.

int main()

{

int arr[3] = { 5, 10, 15 };

int\* ptr = arr;

\*ptr = 30; // set arr[0] to 30

ptr[1] = 20; // set arr[1] to 20

ptr += 2;

ptr[0] = 10; // set arr[2] to 10

ptr -= 2;

while (ptr < &arr[3])

{

cout << \*ptr << endl; // print values

ptr++;

}

}

b.

The function will not work correctly because we are trying to pass an uninitialized pointer to the function. The function parameter is a pointer pToMax, but since ptr is not pointing to anything, there is an error. Instead, we set pToMax to be a pointer to the reference value. This allows us to pass ptr to findMax and change the value of ptr.

void findMax(int arr[], int n, int\* pToMax)

CHANGES TO

void findMax(int arr[], int n, int\*& pToMax)

c.

The variable ptr is not pointing to anything, but the program is attempting to change the variable it is pointing to. We can create a variable that ptr points to in the main function.

void computeCube(int n, int\* ncubed)

{

\*ncubed = n \* n \* n;

}

int main()

{

int cubed = 0;

int\* ptr = &cubed;

computeCube(5, ptr);

cout << "Five cubed is " << \*ptr << endl;

}

d.

The main problem is that the function compares the addresses of str1 and str2 rather than the characters in the C string. To fix this, I changed the comparisons, while loop, and return values to check the character where the pointer is pointing to rather than the address of the pointer. We end the while loop when a value is '\0'.

bool strequal(const char str1[], const char str2[])

{

while (\*str1 != '\0' && \*str2 != '\0')

{

if (\*str1 != \*str2) // compare corresponding characters

return false;

str1++; // advance to the next character

str2++;

}

return \*str1 == \*str2; // both ended at same time?

}

e.

The problem with the program is that the anArray array only exists within the getPtrToArray function. Even though it is returning the address where the array would have started, the array does not exist within the main function, so it will not print 5 4 3 2 1.

2.

a.

double\* cat;

b.

double mouse[5];

c.

cat = &mouse[4];

d.

\*cat = 17;

e.

\*(mouse + 3) = 42;

f.

cat -= 3;

g.

cat[1] = 33;

h.

cat[0] = 25;

i.

bool\* b = new bool(cat[0]==cat[1]);

j.

bool d = (cat == &mouse[0]);

3.

a.

double mean(const double\* scores, int numScores)

{

const double\* ptr = scores;

double tot = 0;

for(int i = 0; i < numScores; i++)

{

tot += \*(ptr + i);

}

return tot/numScores;

}

b.

const char\* findTheChar(const char\* str, char chr)

{

for (int k = 0; \*(str + k) != '\0'; k++)

if (\*(str + k) == chr)

return &\*(str + k);

return NULL;

}

c.

const char\* findTheChar(const char\* str, char chr)

{

while(\*str != '\0')

{

if (\*str == chr)

return &\*str;

str++;

}

return NULL;

}

4.

int main()

{

int array[6] = { 5, 3, 4, 17, 22, 19 };

int\* ptr = maxwell(array, &array[2]); //Turns 5 into -1

\*ptr = -1;

ptr += 2; //Sets ptr to array[2]

ptr[1] = 9; //Turns 17 to 9

\*(array+1) = 79; //Turns 3 to 79

cout << &array[5] - ptr << endl; //Prints 5 - ptr = 3

swap1(&array[0], &array[1]); //array[0] points to array[1] and //array [1] points to array 0

swap2(array, &array[2]); //array [0] switches values with //array[2](Switch -1 and 4)

for (int i = 0; i < 6; i++)

cout << array[i] << endl;

}

It will print:

3 //&array[5] - ptr = 5 - ptr = 5-2

4 //The swaps with array[0] during the swap2 function

79 //\*(array + 1) = array[1], so \*(array+1) = 79 sets array[1] = 79

-1 //maxwell changed the value to -1 and swap2 switched its spot

9 //When ptr was set to 2, ptr[1] refers to array[3], so array[3] = 9

22 //Unchanged

19 //Unchanged

5.

void removeS(char\* message)

{

char\* ptr;

ptr = message;

while (\*ptr != '\0') // '\0' is the string-terminator character

{

while(\*ptr == 's' || \*ptr == 'S')

{

ptr++;

}

\*message = \*ptr;

ptr++;

message++;

}

\*message = \*ptr;

}