Jinkyu Kim

204826628

CS31

Project 6

**1a)**

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

\*(ptr + 2) = 10; // set arr[2] to 10

while (\*ptr >= arr[2])

{

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

ptr++;

}

}

**1b)**

Problem: since ptr is not declared in the main function, it cannot be copied into the function findMax. Thus, it requires pass by pointer-reference so that I can be modified within the function.

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

{

if (n <= 0)

return; // no items, no maximum!

pToMax = arr;

for (int i = 1; i < n; i++)

{

if (arr[i] > \*pToMax)

pToMax = arr + i;

}

}

int main()

{

int nums[4] = { 5, 3, 15, 6 };

int\* ptr;

findMax(nums, 4, ptr);

cout << "The maximum is at address " << ptr << endl;

cout << "It's at position " << ptr - nums << endl;

cout << "Its value is " << \*ptr << endl;

}

**1c)**

Problem: In the main function, ptr is not initialized. In addition, even if ptr was initialized, ptr will simply be copied into the function computeCube. Therefore, it needs a pass by reference so that what happens in computeCube can be stored.

void computeCube(int n, int\* ncubed)

{

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

}

int main()

{

int k;

int\* ptr = &k;

computeCube(5, ptr);

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

}

**1d)**

Problem: This function compares all characters in str1 and str2 at once. They need to be compared individually using pointers.

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 true; // both ended at same time? }

int main()

{

char a[15] = "Shen";

char b[15] = "Shi";

if (strequal(a, b))

cout << "They're the same person!\n";

}

**1e)**

Problem: ptr is pointing to a local variable that exists only in getPtrToArray. Upon returning the value, it disappears and ptr ends up pointing to an array that does not exist

**2.**

a. double\* cat;

b. double mouse[5];

c. cat = &mouse[4];

d. \*cat = 25;

e. \*(mouse + 3) = 54;

f. cat -= 3;

g. cat[1] = 42;

h. cat[0] = 27;

i. bool b = (\*cat == \*(cat + 1));

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

**3a)**

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

{

const double\* ptr = scores;

double tot = 0;

int k = 0;

while (ptr+k != scores + numScores) // &scores[0] + numScores

{

tot += \*(ptr + k);

k++;

}

return tot/numScores;

}

**3b)**

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

{

const char\* ptr = str; // \*ptr = str[0]

for (int k = 0; \*(ptr + k) != 0; k++) //ptr = &str[0], ptr + k = &str[k]

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

return (ptr + k);

return nullptr;

}

**3c)**

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

{

const char\* ptr = str; // \*ptr = str[0]

while (\*ptr != 0)

{

if (\*ptr == chr)

return ptr;

ptr++;

}

return nullptr;

}

**4)**

int\* maxwell(int\* a, int\* b)

{

if (\*a > \* b)

return a;

else

return b;

}

void swap1(int\* a, int\* b)

{

int\* temp = a;

a = b;

b = temp;

}

void swap2(int\* a, int\* b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main()

{

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

//array is equal to array[0], which is 5.

//&array[2] is equal to array[2], which is 4.

//Thus, maxwell returns a pointer to the array, and ptr points to array[0], which is 5.

int\* ptr = maxwell(array, &array[2]);

// array[0] becomes -1

\*ptr = -1;

// ptr now points to array[2]

ptr += 2;

// ptr[1] is equal to ptr[3], and thus array[3] becomes 9

ptr[1] = 9;

// array[1] turns to 79

// the array has become [-1, 79, 4, 9, 22, 19]

\*(array + 1) = 79;

//This is equal to &array[5] - &array[2]. Since array[5] is located 3 right to array[2], this will print 3.

cout << &array[5] - ptr << endl;

//swap1 only swaps the address of array[0] and array[1], not the object.

//Thus, the object remains the same

swap1(&array[0], &array[1]);

//swap2 swpas the object of array[0] and array[2].

//Thus, the array now becomes [4, 79, -1, 9, 22, 19]

swap2(array, &array[2]);

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

cout << array[i] << endl;

//This will print :

3

4

79

-1

9

22

19

}

**5)**

void removeS(char\* msg)

{

char\* ptr = msg;

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

{

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

{

ptr++;

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

continue;

}

\*msg = \*ptr;

msg++;

ptr++;

}

\*msg = '\0';

}