**CSCD 240**

NOTE: Your answers, for all problems, will be saved in a file named cscd240Lab6pointers.pdf for all problems

NOTE: Your C file will be named cscd240Lab6.c

1. Type in, compile and execute the following code.

#include <stdio.h>

int main()

{

int arr[] = { 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000};

int \*ptr = arr;

/\* This gives us an idea of the memory map \*/

printf("sizeof(ptr) %ld\n", sizeof(ptr) ); // number of bytes used for a pointer variable

printf("sizeof(arr[0]) %ld\n", sizeof(arr[0]) ); //number of bytes used for an array element

printf("value of arr is %p\n", arr);

printf("value of ptr is %p\n", ptr);

printf("address of arr[1] is %p\n", &arr[1]);

printf("addres of arr[9] is %p\n", &arr[9]);

printf("&ptr is %p\n", &ptr);

/\* end memory map \*/

return 0;

}// end main

This code will provide a base address for arr and ptr.

For this problem use the **presumed** base address of 0x5600bc

Answer /complete the following

* 1. What is the size of ptr on cslinux machine?
  2. What is the size of arr[0] on cslinux machine?
  3. Draw a memory map that shows the memory locations of each element of the array and of ptr. Please follow the patterns that we used in the classroom, with values shown in memory cell (box) and address shown outside of memory cell.

1. Use the **presumed** base address **0x5600bc** provided from problem #1 as the base address of the array. Based on the code below, create an educated guess that clearly outlines what you believe will happen as each line is executed. In your explanation clearly **explain what is happening**, don't just give memory addresses or values. If you only provide memory addresses or values you will receive **0 points** for this problem. Your guesses will be clearly labeled in the PDF file. If necessary, please refer to C operators **precedence table regarding which operator is evaluated first or next**. You must provide the line of code and then the explanation. An example is provided in below.

**(For example**) ptr++

**Explain:** ptr is a pointer, currently (before ptr ++ is executed) points to array element X. After ptr++ is executed, ptr points to array element Y because pointer is increased by one, thus pointing to the next element in the array. My guess is right.

**Note:** the code provided in the below is code segment, not a complete program. Please define your main() and include required header files.

**Note**: Where I provide the comment, there is no explanation required.

int arr[] = { 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000}; //No need to explain this

int \*ptr = arr; //No need to explain this statement.

ptr++;

printf("\*ptr %d\n", \*ptr);

printf("ptr %p\n", ptr);

\*++ptr;

printf("\*++ptr %d\n", \*ptr);

printf("ptr %p\n", ptr);

\*ptr++;

printf("\*ptr++ %d\n", \*ptr);

printf("ptr %p\n", ptr);

ptr = arr; // reset ptr, no need to explain this statement

// fun with printf repeat last couple of commands

printf("\*++ptr %d\n", \*++ptr);

printf("ptr %p\n", ptr);

printf("\*ptr++ %d\n", \*ptr++);

printf("ptr %p\n", ptr);

ptr = arr; // reset ptr, no need to explain this statement

\*ptr += 1;

printf("\*ptr %d\n", \*ptr);

printf("ptr %p\n", ptr);

printf("\*(ptr+1) = %d\n", \*(ptr+1));

ptr = arr; // reset ptr, no need to explain this statement

\*(arr+2) = \*ptr+100;

printf("\*(arr+2) = %d\n", \*(arr+2));

ptr = arr + 5;

printf("\*ptr %d\n", \*ptr);

printf("ptr %p\n", ptr);

ptr = arr; // reset ptr, no need to explain this statement

arr[2] = \*(ptr + 5);

printf("arr[2] = %d\n", arr[2]);

ptr = (arr + 10);

printf("ptr %p\n", ptr);

printf("\*ptr %d\n", \*ptr);

1. Edit the C file
   1. Add the code from problem #2 to your C file
   2. Compile and execute your C file - capture the output with one or more screen shot(s).
   3. In the PDF clearly state the line of code, your guess and what the result was. If you guessed correctly then state – correct guess, otherwise clearly explain the incorrect guess.

**TO TURN IN:**

A zip file containing:

* Your PDF file
* Your C file

Please name your files with our naming scheme.