

CSCD 240

NOTE: Your answers, for all problems, will be saved in a file named cscd240Lab8pointers.pdf for all problems

NOTE: Your C file will be named cscd240Lab8.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) );
    printf("sizeof(arr[0]) %ld\n", sizeof(arr[0]) );

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

    printf("arr[1] %p\n", &arr[1]);
    printf("arr[9] %p\n", &arr[9]);
    printf("&ptr %p\n", &ptr);
    /* end memory map */
    return 0;

} // end main
```

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

Use the address of 0x7fff3d830280 for the starting location of the array.

Use the address of 0x7fff3d8302a8 for the location of ptr.

Answer /complete the following

- What is the size of ptr?
- What is the size of arr[0]?
- Draw a memory map that shows the memory locations of each element of the array and of ptr.

- 2) Use the provided address from #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. You must provide the line of code and then the explanation. NOTE: Where I provide the comment // reset ptr no explanation is required

```
ptr++;  
printf("*ptr %i\n", *ptr);  
printf("ptr %p\n", ptr);
```

```
*++ptr;  
printf("*++ptr %i\n", *ptr);  
printf("ptr %p\n", ptr);
```

```
*ptr++;  
printf("*ptr++ %i\n", *ptr);  
printf("ptr %p\n", ptr);
```

```
ptr = arr; // reset ptr
```

```
// fun with printf repeat last couple of commands  
printf("*++ptr %i\n", *++ptr);  
printf("ptr %p\n", ptr);
```

```
printf("*ptr++ %i\n", *ptr++);  
printf("ptr %p\n", ptr);
```

```
ptr = arr; // reset ptr
```

```
*ptr += 1;  
printf("*ptr %i\n", *ptr);  
printf("ptr %p\n", ptr);
```

```
printf("*(ptr+1) %i\n", *(ptr+1));
```

```
ptr = arr; // reset ptr
```

```
*(arr+2) = *ptr+100;
printf("*(arr+2) %i\n", *(arr+2));
```

```
ptr = arr + 5;
printf("*ptr %i\n", *ptr);
printf("ptr %p\n", ptr);
```

```
ptr = arr; // reset ptr
```

```
arr[2] = *(ptr + 5);
printf("arr[2] %i\n", arr[2]);
```

```
ptr = (arr + 10);
printf("ptr %p\n", ptr);
printf("*ptr %i\n", *ptr);
```

3) Edit the C file

- a) Add the code from problem #2 to your C file
- b) Compile and execute your C file - capture the output
- c) 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.
- d) Explain how the value for *ptr was determined based on

```
ptr = (arr + 10);
printf("ptr %p\n", ptr);
printf("*ptr %i\n", *ptr);
```

TO TURN IN:

A zip file containing:

- Your PDF file
- Your C file

You better know the naming scheme.