# Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ EECE 2160 Summer 2016

**Pointers and Arrays**

The purpose of this exercise is to help you understand how pointers work. Try to study and understand what the output is, do not just print it out.

**Exercise 0 – Main function**

Use the following main function to test your functions:

|  |
| --- |
| #include <stdio.h>  int main(){  // uncomment the appropriate line to test your code for the exercises below    //Exercise1();  //Exercise2();  //Exercise3();  //Exercise4();  //Exercise5();  return 0;  } |

Each exercise below should be done in a separate function. For example, Exercise 1 should be programmed in the following format:

void Exercise1() {

// Exercise1 implemented here

}

In the **main** function (from Exercise 1 above), call this function once via:

Exercise1();

**Exercise 1**: Introduce int variables x, y, z and int\* pointer variables p, q, r. Set x, y, z to three distinct values. Set p, q, r to the addresses of x, y, z respectively.

1. Print with labels the values of x, y, z, p, q, r, \*p, \*q, \*r. (use these lines of code)

printf("x = %d, p = %X, \*p = %d\n", x, p, \*p);

printf("y = %d, q = %X, \*q = %d\n", y, q, \*q);

printf("z = %d, r = %X, \*r = %d\n", z, r, \*r);

(2) Print the message: Swapping values.

(3) Execute the swap code: z = x; x = y; y = z;

(4) Print with labels the values of x, y, z, p, q, r, \*p, \*q, \*r. (use the lines of code above)

(5) Study and explain the output.

**Exercise 2**: Introduce int variables x, y, z and int\* pointer variables p, q, r. Set x, y, z to three distinct values. Set p, q, r to the addresses of x, y, z respectively.

(1) Print with labels the values of x, y, z, p, q, r, \*p, \*q, \*r. (use the lines of code above)

(2) Print the message: Swapping pointers.

(3) Execute the swap code: r = p; p = q; q = r;

(4) Print with labels the values of x, y, z, p, q, r, \*p, \*q, \*r. (use the lines of code above)

(5) Study and explain the output.

**Exercise 3**: Define an array int arr[5]:

Then:

(1) Write a loop to fill the array arr with 21, 22, 23, 24, 25.

(2) Execute the following loop:

for (i = -2; i < 7; i++)

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

(3) Study and explain the output.

**Exercise 4**: Define an array int arr[5] and fill this array with values 1, 4, 7, 10, 13. Introduce an int i and an int\* pointer variable p. Then run two printing loops and explain the output:

for (i = 0; i < 5; i++)

printf("index %d " " %d " " %d\n", i, \*(arr+i), arr[i]);

printf("\n");

i = 0;

p = arr;

while (p < (arr+5)) {

printf("%d " " %x " " %d\n", i, p, \*p);

i++;

p++;

}

Exercise 5: Define an **int\*** pointer variable **arr**. Then:

(1) Use malloc to make arr point to a dynamic array of 5 cells of type int.

(2) Write a loop to fill arr with values 3, 7, 11, 15, 19.

(3) Using Hex format, print the pointer address stored in arr.

(4) Write a loop to print the values in arr with one cell per line.

(5) Delete the dynamic memory allocated to arr using free()

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Summer 2016 EECE2160 J. Kimani

**Pointers and Arrays Exercise 1**

1. Exercise 1:

|  |
| --- |
|  |

1. Exercise 2:

|  |
| --- |
|  |

1. Exercise 3:

|  |
| --- |
|  |

1. Exercise 4:

|  |
| --- |
|  |

1. Exercise 5:

|  |
| --- |
|  |

1. Attach your final C program here (you can copy and paste it):

|  |
| --- |
|  |