**Tutorial - 8**

**Objectives:** To practice with

* Pointers

1. **Assuming the following declaration of the variables** cType **and** voltage. **Declare the corresponding number of pointer type variables and assign them with the memory addresses of** cType **and** voltage.

Char cType;

float voltage;

1. **Can one pointer be used to access several variables (one at a time) of the same type? Provide an example.**
2. **Provide an example of changing a variable value using a pointer and the indirection operator**
3. **What are the values of ptrI and ptrC?**

int \*ptrI = 1000;

char \*ptrC = 1000;

ptrI += 2;

ptrC += 2;

**Explain the results.**

# Given the declarations:

int m = 25, n = 77;

char c = '\*';

int \*itemp;

# describe the errors in each of the following statements.

m = &n;

itemp = m;

\*itemp = &c;

1. **Write a program in C to find the factorial of a given number using pointers.**
2. **a) What are the values of main function variables x and y at the point marked**

**/ \* values here \*/ in the following program?**

/\* nonsense \*/

void silly(int x);

int

main(void)

{

int x, y;

x = 10; y = 11;

silly(x);

silly(y); /\* values here \*/

. . .

}

void

silly(int x)

{

int y;

y = x + 2;

x \*= 2;

}

**b)**

/\* nonsense \*/

void silly(int \*x);

int

main(void)

{

int x, y;

x = 10; y = 11;

silly(&x);

silly(&y); /\* values here \*/

. . .

}

void

silly(int \*x)

{

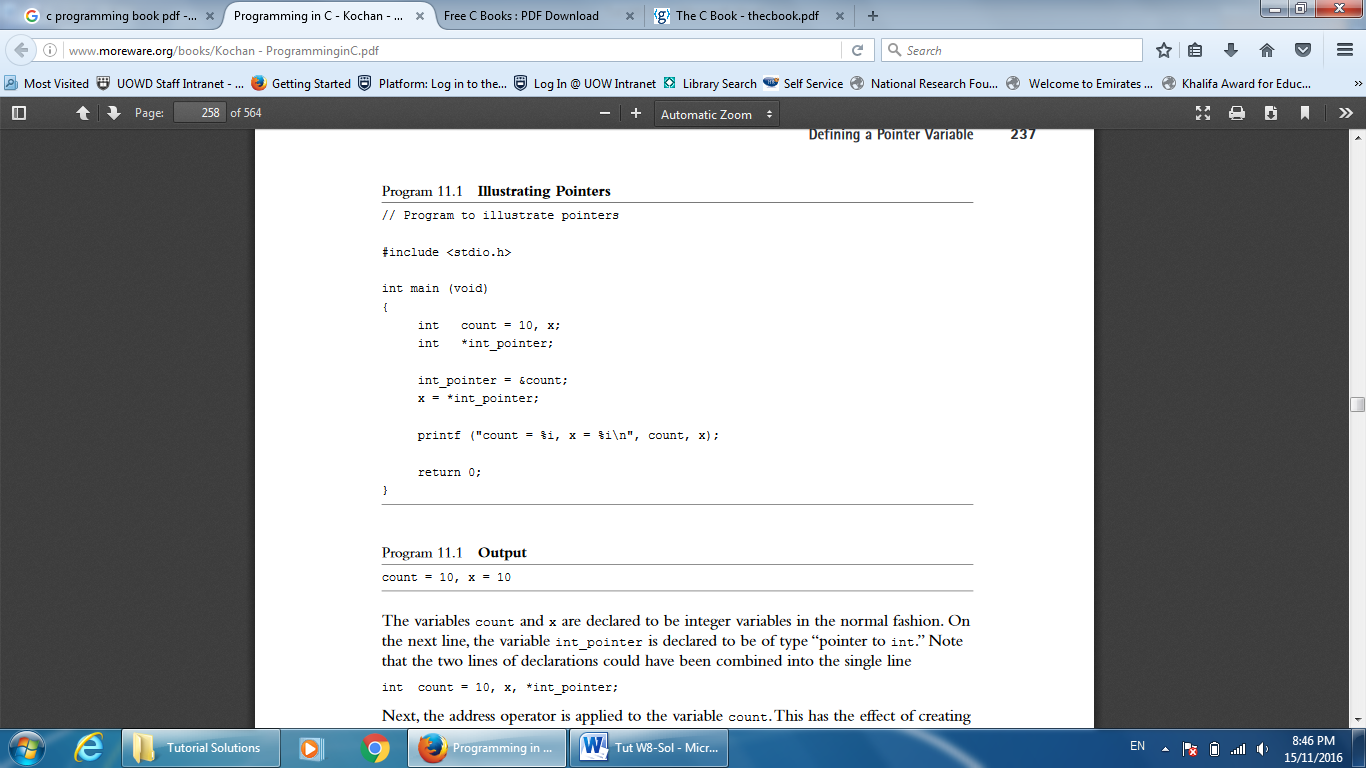
int y;

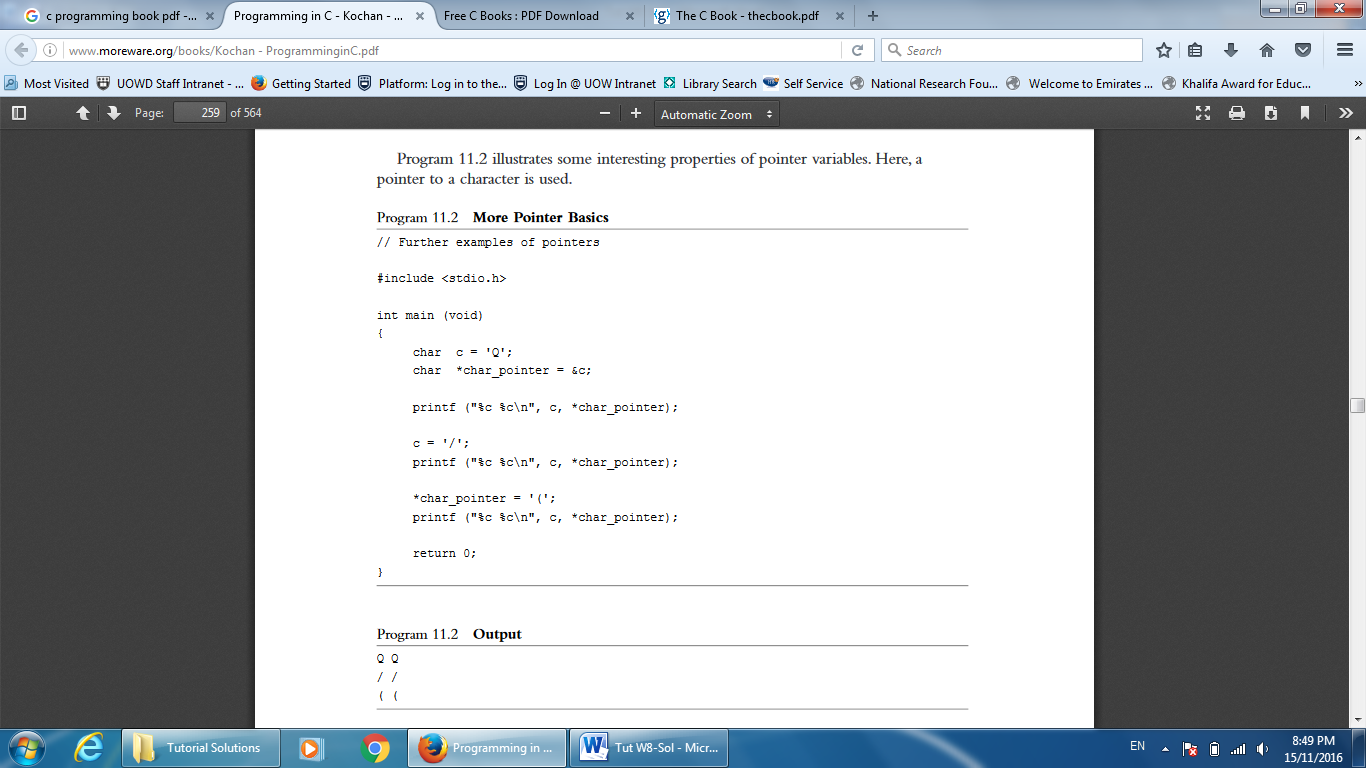
y = \*x + 2;

\*x = 2 \* \*x;

}

1. **What is the output of the following programs?**





1. **Write a statement to execute the following:**
   1. **Declare a pointer to an integer called address.**
   2. **Assign the address of a float variable** balance **to the float pointer** temp**.**
   3. **Assign the character value 'W' to the variable pointed to by the char pointer** letter***.***
   4. **Declare a pointer to the text string "Hello" called** message**.**
   5. **Assume float balance[10][5]. How can you access balance[3][1] using pointers?**
2. **Allocate memory to store an array of 50 integers and initialize elements of this dynamic array with values equal to their indexes (0 – 49). The code must provide memory allocation error checking.**
3. **Assuming that the function** getNextValue() **correctly returns a** float **type value, what is wrong with this code and fix it?**

int i;

float \*flArr;

if( (flArr = calloc(256, sizeof(float))) == NULL)

{

fprintf(stderr, " Memory allocation failure"); return (-1);

}

for(i=0; i<256; i++)

\*flArr++ = getNextValue()**;**

free( flArr );

1. **Define a function**

char \*createEmptyString( int length);

**that creates an empty c-string of a specified length. Write a simple main()function to test createEmptyString().**

# Write a sample of code to allocate memory for:

* **one item of type component and initialize it**
* **A component consists of char type[7]; float price; int quantity;**
* **a dynamic array with SIZE elements of type component and initialize the 3rd element.**

1. **Differentiate between dynamic and nondynamic data structures.**