**Spring 2020 EEL2880: Software Techniques: Program 1: Hello and Escape Characters**

**Dr. Subbarao V. Wunnava: Ref: Chp 1, 2, 3 &4 Deitel: Modified 01 14 2020**

// EEL 2880 Engineering Software Techniques; Copy right extended thru 2020

// Summer 2015 05 14 2015: Modified Spring 2017 Subbarao; hello\_.c C PROGRAM 1; Ref: Deitel Chp 1 & 2 3&4

// First program with C, demonstrating Escape Characters /n: New line; /t :tab; /a: bell-

#include <stdio.h> // preprocessor directives needed to get the program flow

#include <stdlib.h>// for I/O (Input/Output) and processing operations

int main(void) // start of the main program

{

int n = 0, t = 0, a = 0; // integer variables for new line, tab, and bell escape characters

printf(" \n\n\t Pr1: Hello world! EEL 2880 Eng Software Tech Subbarao 05 14 2015 \a\n");

printf(" \a Hello C program in IDE 2010 Visual Studio Environment \n\n");

n = n+5;

t = t+1;

a = a+2;

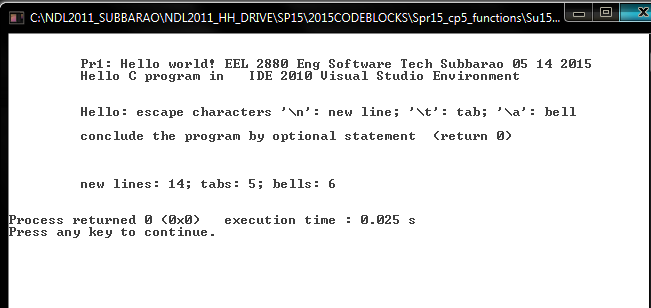
printf(" \n\t\a Hello: escape characters '\\n': new line; '\\t': tab; '\\a': bell");

printf(" \n\n\t\a conclude the program by optional statement (return 0) \n\n");

printf(" \n\n\t\a new lines: %d; tabs: %d; bells: %d \n\n\t\a", (n+9), (t+4), (a+4));

return 0;

} // conclusion of the main program



**Variable Declarations**: For efficient and appropriate mathematical operations, variables should be properly declared. Integer operations only deal with integers and provide integer results. Decimal or floating point declarations provide decimal operations. There are several other types of variables, each type has a definite specification.

**Spring 2020 : EEL 2880: Software Pr2 Modified: 01 14 2020**

**Subbarao Wunnava Add/Subtract/Multiply/Divide/Compare: Pr2**

//01\_14\_2020\_AddSubtractMultiplyDivide& Relational C Program Subbarao Wunnava Pr2

//Also relational operators specified ref: Deitel chp 2 & 3 pgs 55 -77: Modified 01 12 2017: Spring 2020

#include <stdio.h> // pre-processor directives

#include <stdlib.h>

#include <math.h>

int main (void) // start with 'main'. void used to specify self-continued program

// declare variables both integer and floating point

{

int num1, num2, num3, num4; // integer variables

int sum12, sum34, dif12, dif34, prod12, prod34, idiv12, idiv34; // integer variables

float fdiv12, fdiv34; // floating point (decimal) variables

printf("\n 05\_14\_2015\_AddSubtractMultiplyDivide& Relational: Subbarao Wunnava: C-Pr2 \n\n");

// read 4 numbers num1, num2, num3, num4

printf (" \t Enter 4 numbers num1, num2, num3, num4 \a\a \n");

scanf("%d %d %d %d", &num1, &num2, &num3, &num4); // %d specifier for integers

//perform arithmetic operations

sum12 = (num1 + num2);

sum34 = (num3 + num4);

dif12 = (num1 - num2);

dif34 = (num3 - num4);

prod12 = (num1\*num2);

prod34 = (num3\*num4);

idiv12 = (num1/num2);

idiv34 = (num3/num4);

fdiv12 = (float) num1/num2;// floating point or decimal operation conducted

fdiv34 = (float) num3/num4;

// outputting the results

printf (" numbers num1, num2, num3, num4 are:\a\a %d %d %d %d \n\n", num1, num2, num3, num4);

printf (" sum12, sum34 dif12, dif34 are:\a\a %d %d %d %d \t\n\n", sum12, sum34, dif12, dif34);

printf (" prod12, prod34, idiv12, idiv34 are: %d %d %d %d \t\n\n", prod12, prod34, idiv12, idiv34);

printf ("\a\a fdiv12 and fdiv34 are: %.2f %.4f \t\t \n ", fdiv12, fdiv34);

printf ("\a\a\n");

// compare operations and decision making with if loops

if (sum12 >= sum34)

{

printf (" sum12 is larger than or equal to sum34 \a\n\n");

} // end first if loop

if (sum34 > sum12)

{

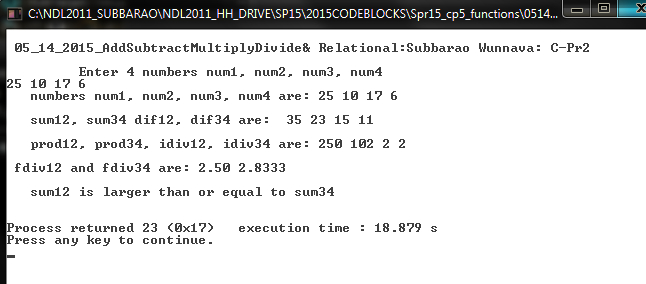
printf (" sum34 is larger than sum12 \a\n\n");

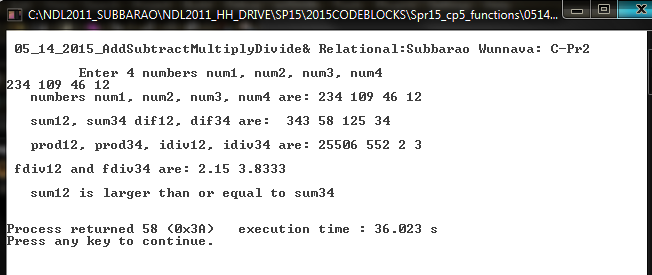
} // end second if loop

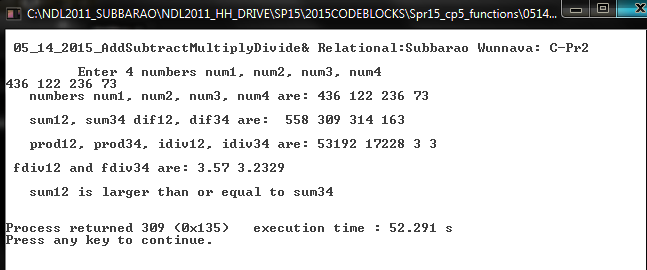
// system ("PAUSE"); for Microsoft software modules, system (“PAUSE”) needed to stabilize display

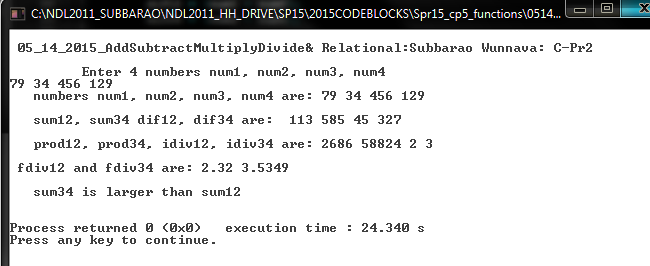
// return 0; for Microsoft software modules, return 0 may be needed for successful program termination

} //end of the main program

****

****

****

****

**2.12** What, if anything, prints when each of the following statements is performed? If nothing

prints, then answer “Nothing.” Assume x = 2 and y = 3.

a) printf( **"%d"**, x );

**ANS:** 2

b) printf( **"%d"**, x + x );

**ANS:** 4

c) printf( **"x="** );

**ANS:** x=

d) printf( **"x=%d"**, x );

**ANS:** x=2

e) printf( **"%d = %d"**, x + y, y + x );

**ANS:** 5 = 5

f) z = x + y;

**ANS:** Nothing. Value of x + y is assigned to z.

g) scanf( **"%d%d"**, &x, &y );

**ANS:** Nothing. Two integer values are read into the location of x and the location of y.

h) // printf( "x + y = %d", x + y );

**ANS:** Nothing. This is a comment.

i) printf( **"\n"** );

**ANS:** A newline character is printed, and the cursor is positioned at the beginning of the

next line on the screen.

**2.14 Given the equation *y = ax3 +* 7, which of the following, if any, are correct C statements for**

**this equation?**

**a) y = a \* x \* x \* x + 7;**

**b) y = a \* x \* x \* ( x + 7 );**

**c) y = ( a \* x ) \* x \* ( x + 7 );**

**d) y = ( a \* x ) \* x \* x + 7;**

**e) y = a \* ( x \* x \* x ) + 7;**

**f) y = a \* x \* ( x \* x + 7 );**

**ANS: a, d, and e.**