**Lab\_09: Nesting Loops**

Name: Muhammad Rizwan Khalid

Class: BSCS-6A

Reg.No: 180459

**Task: 02: Nesting Loops: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

// this is #1 - I'll call it "CN"

for (int n = 1 ; n <= 3; n++)

{

for (char c = 'A'; c <= 'E'; c++)

{

printf("%d %c\n", n, c);

}

}

printf("\n\n");

// this is #2 - I'll call it "AB"

for (int a = 1; a <= 3; a++)

{

for (int b = 1; b <= 3; b++)

{

printf("%d-%d ", a, b);

}

printf("\n");

/\* your code comes here!!! \*/

}

system("pause");

return EXIT\_SUCCESS;

}

/\*Question: Look at the first set of nested loops ("CN"). Which variable changes faster? Is it the

variable controlled by the outer loop (c) or the variable controlled by the inner loop

(n)?

Answer: Inner Loop (n) changes faster because for one increment of outer there are three increments

for inner loop.

Question: Change the order of the loops so that the "c" loop is on the inside and the "n" loop is on

the outside. How does the output change?

Answer: The output changes in such a way that each number is followed by first five capital alphabets.

Question: Look at the second set of nested loops ("AB"). Add newline (\n) at the end of printf()

statement. How does the output change?

Answer: The output changes in such a way that the horizontal data is converted into a vertical data.

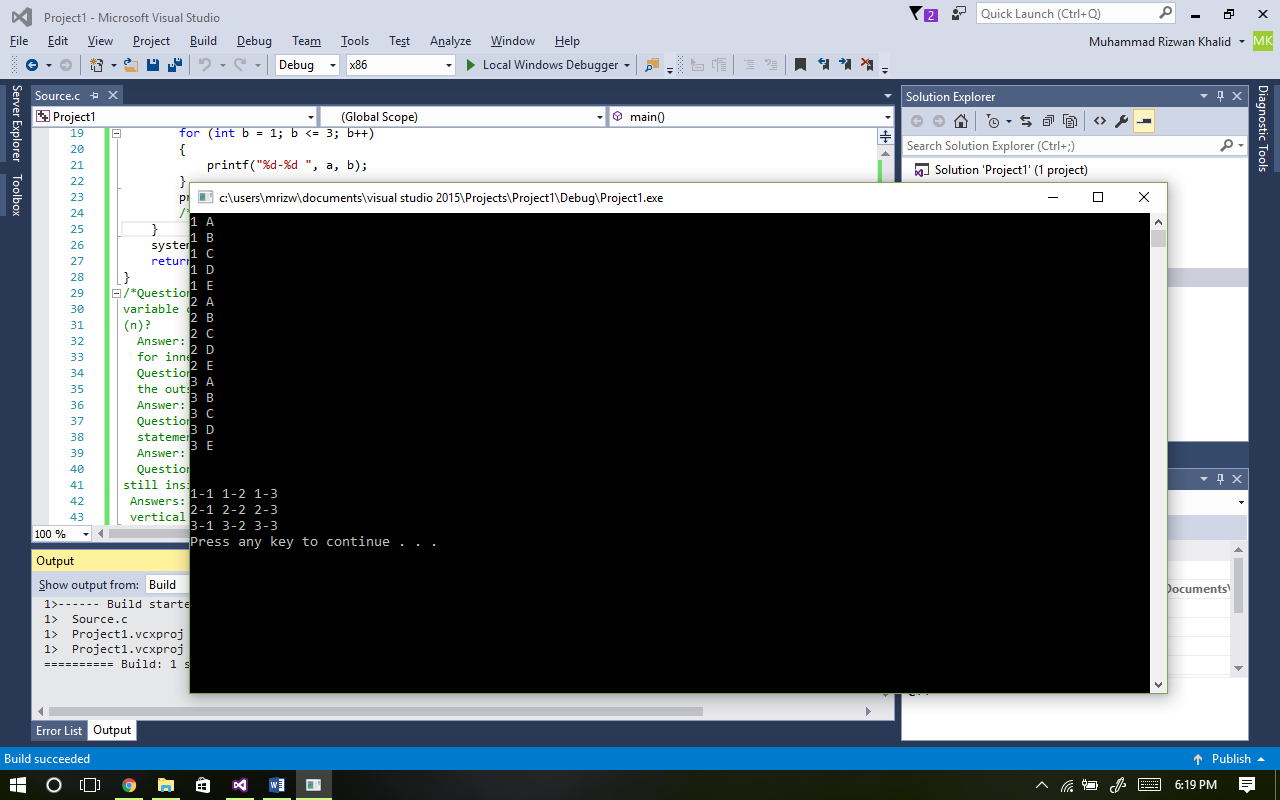
Question: Add a printf("\n") statement after the close brace of the inner loop (the "b" loop), but

still inside the outer loop. How does the output change?

Answers: The output changes in such a way that horizontal data is converted into horizontal rows and

vertical columns\*/

**Output: -**



**Task: 03: Odometer Loops: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void delay(int);

int main()

{

int a;

printf("Which Base (2-10)?");

scanf("%d", &a);

for (int thous = 0; thous<a; thous++)

{

for (int hund = 0; hund<a; hund++)

{

for (int tens = 0; tens<a; tens++)

{

for (int ones = 0; ones<a; ones++)

{

printf(" %d%d%d%d\r", thous, hund, tens, ones);

delay(10);

}

}

}

}

printf("\n");

system("pause");

return EXIT\_SUCCESS;

}

void delay(int milliseconds)

{

long pause;

clock\_t now, then;

pause = milliseconds \* (CLOCKS\_PER\_SEC / 1000);

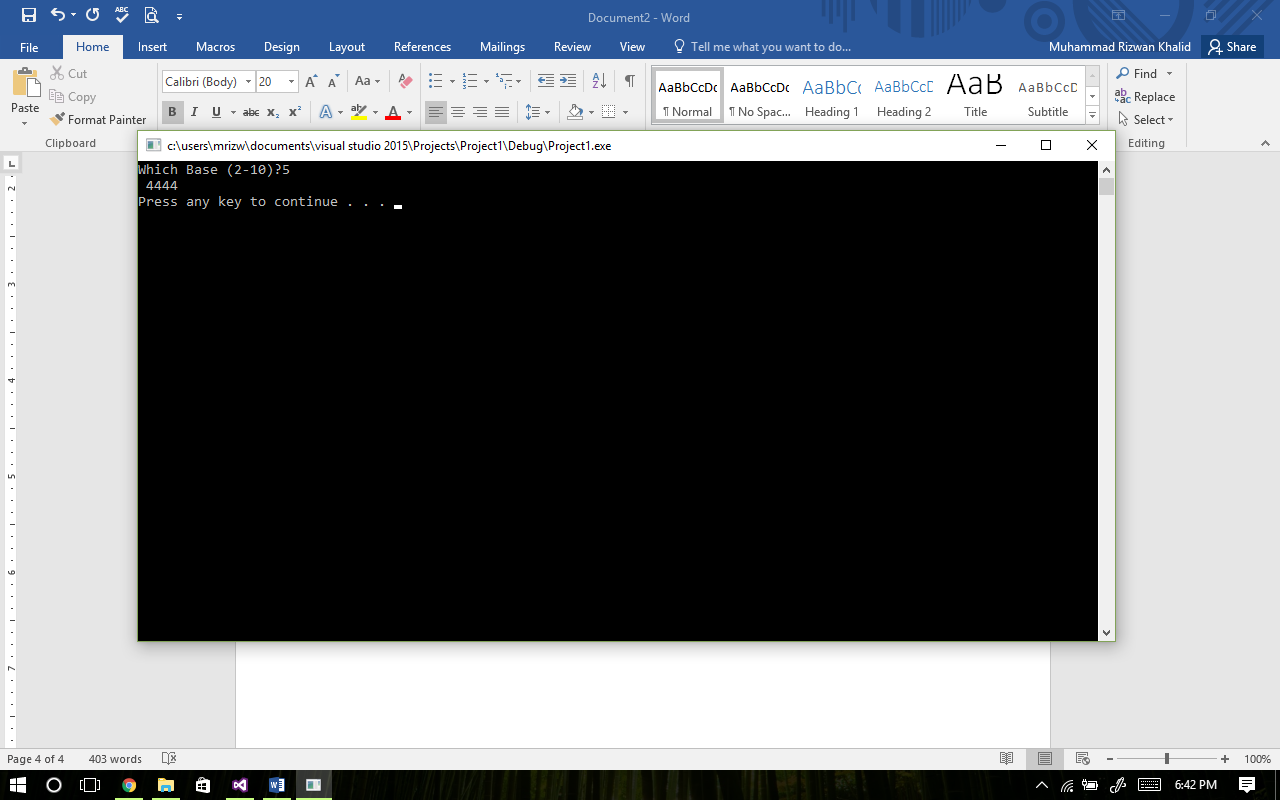
now = then = clock();

while ((now - then) < pause)

now = clock();

}

**Output: -**



**Task: 04: Basic Nested Loops: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

for (int a = 0; a <= 5; a++)

{

for (int b = 0; b <= 5; b++)

{

printf("(%d , %d) ", a, b);

}

printf("\n");

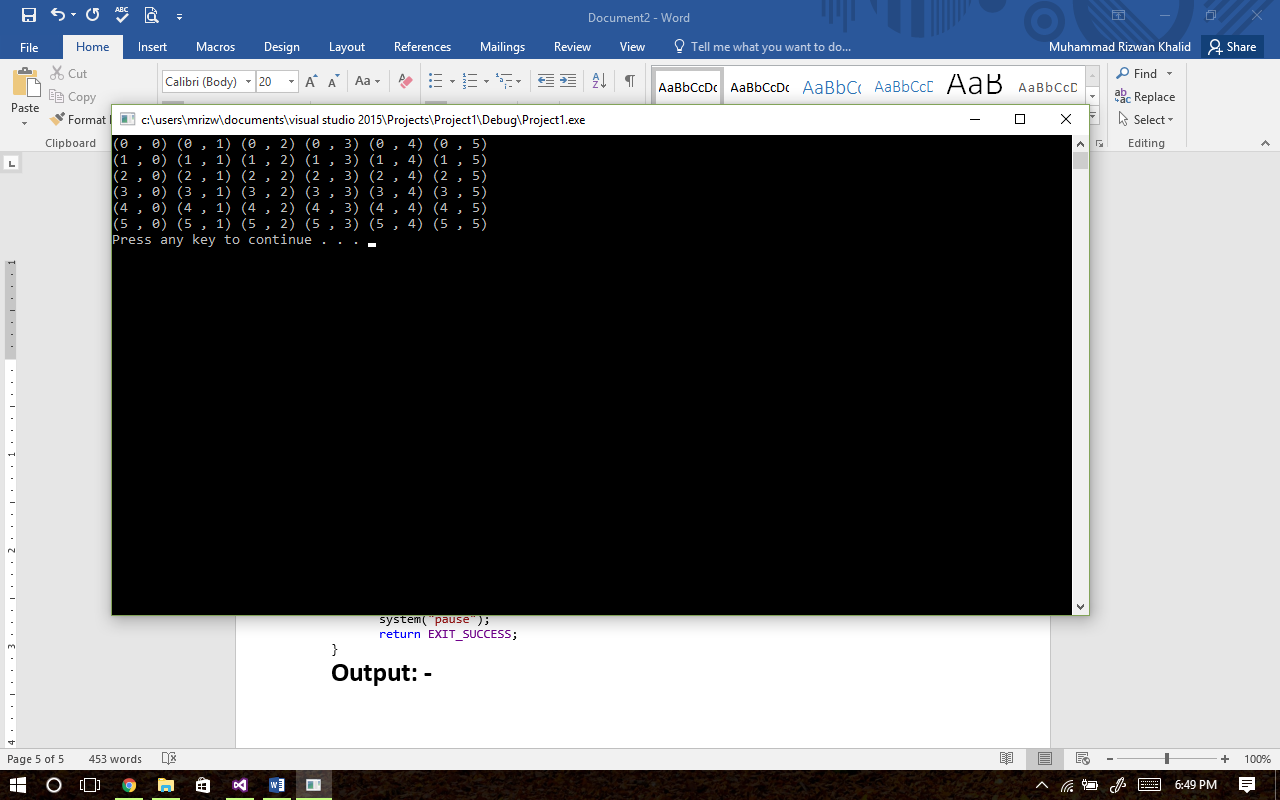
}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**



**Task: 5: Multiplication Table: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf(" x | 1 2 3 4 5 6 7 8 9\n");

for (int a = 1; a <= 12; a++)

{

printf(" %d | ", a);

for (int b = 1; b <= 9; b++)

{

printf(" %d", a\*b);

}

printf("\n");

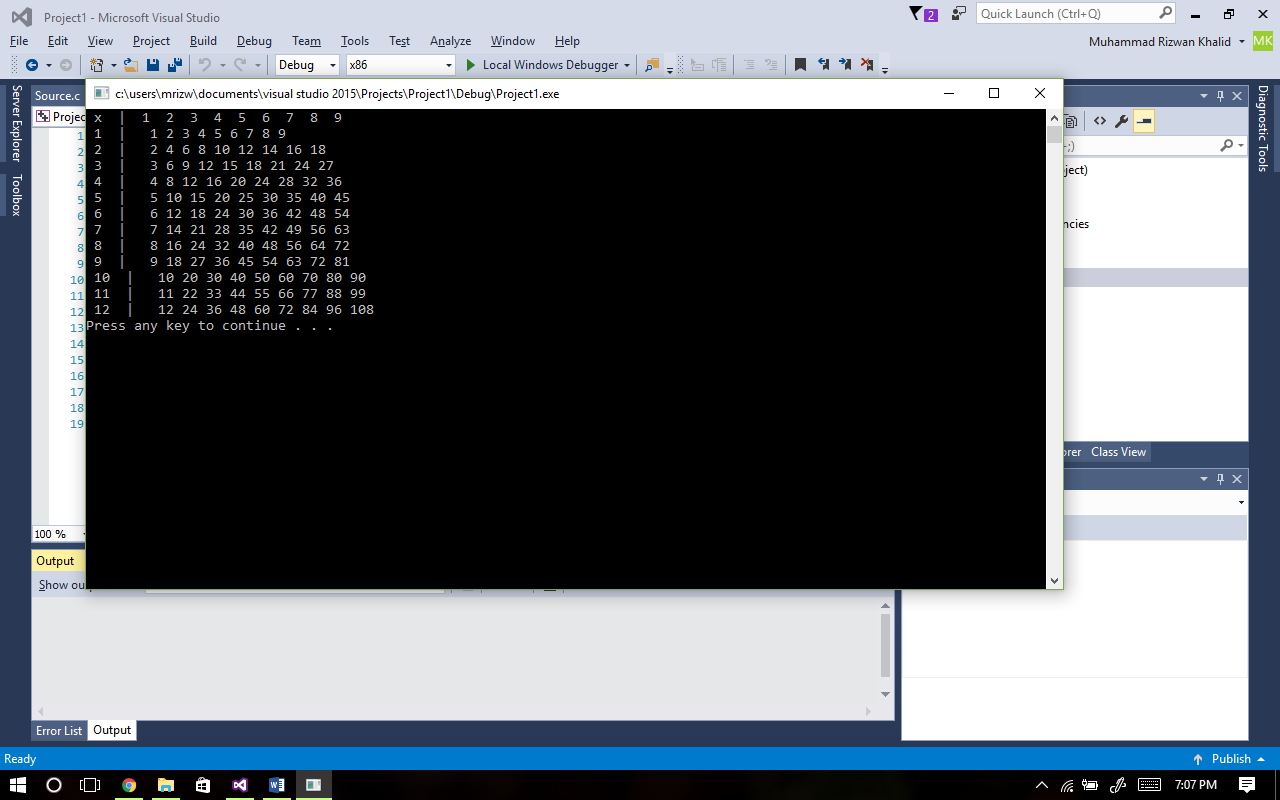
}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**



**Task: 06: Number Puzzles i: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

for (int a = 10; a < 99; a++)

{

for (int b = 0; b < 99; b++)

{

if ((a + b == 60) && ((a - b == 14) || (b - a == 14)))

printf("(%d , %d) ", a, b);

}

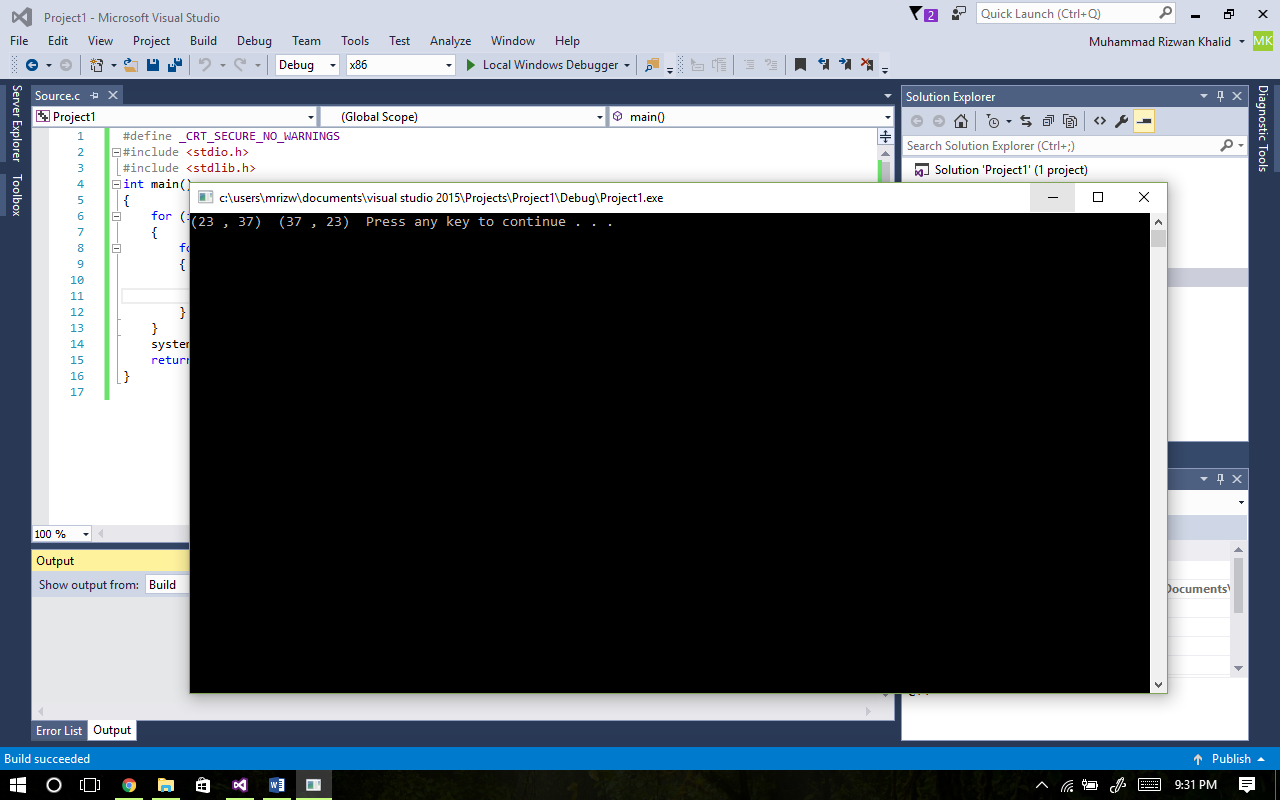
}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**



**Task: 07: Getting individual digits: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

for (int b = 10; b <= 99; b++)

{

printf("%d, %d+%d = %d\n", b, b / 10, b % 10, (b / 10) + (b % 10));

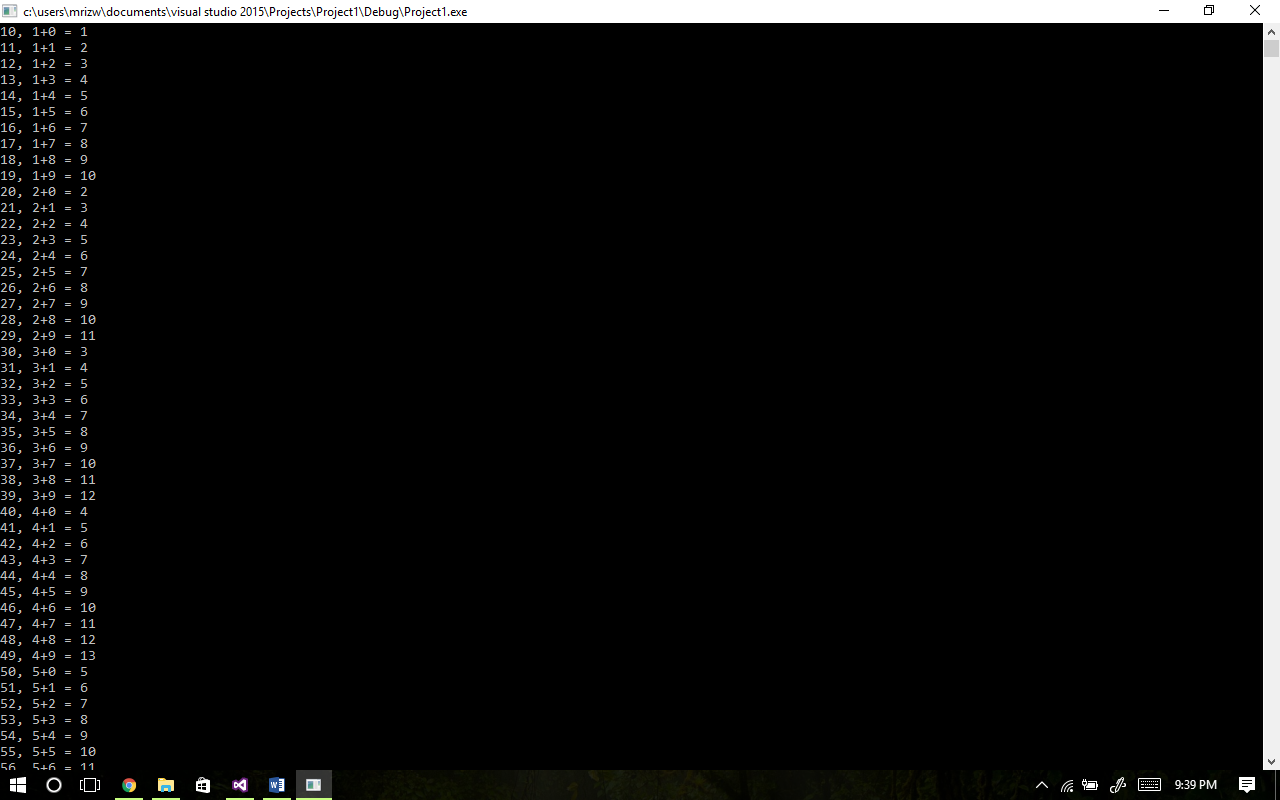
}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**



**Task: 09: Armstrong Numbers: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

for (int num = 100 ; num <= 999 ; num++)

{

int a = num / 100;

int b = (num / 10) % 10;

int c = num % 10;

if (a\*a\*a + b\*b\*b + c\*c\*c == num)

{

printf("(%d)^3 + (%d)^3 + (%d)^3 = %d\n", a, b, c, num);

}

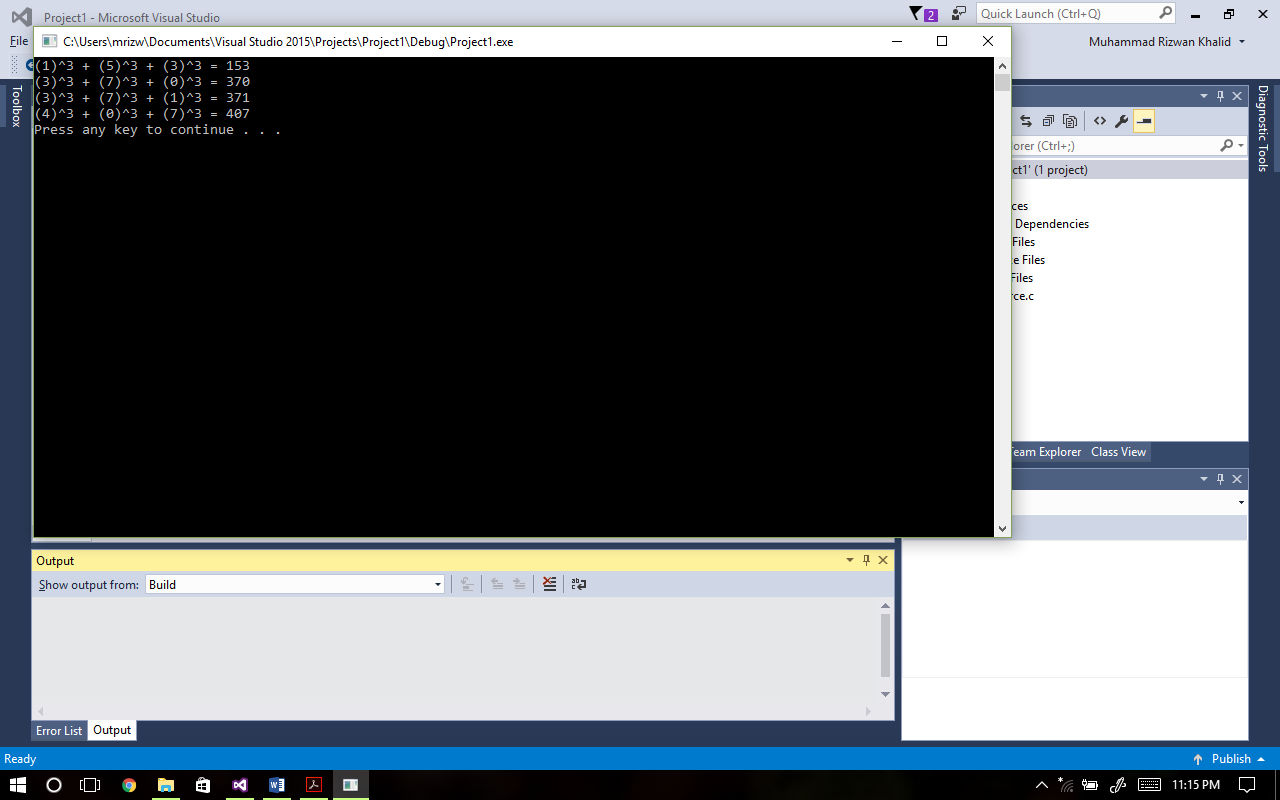
}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**



**Task: 08: More Number Puzzles: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int main()

{

int num, num1, num2;

printf("1) Find two digits numbers <= 56 with sums of digit > 10\n");

printf("2) Find two digit number minus number reversed which equal sum of digits\n");

printf("3) Quit\n\n");

scanf("%d", &num);

switch (num)

{

case 1:

{

for (num1 = 10; num1 <= 99; num1++)

{

if ((num1 <= 56) && (((num1 / 10) + (num1 % 10)) > 10))

printf("\n%d\n", num1);

}

break;

}

case 2:

{

for (num2 = 10; num2 <= 99; num2++)

{

if ((num2 - (((num2 % 10) \* 10) + (num2 / 10))) == ((num2 / 10) + (num2 % 10)))

printf("\n%d\n", num2);

}

break;

}

default:

{

break;

}

}

system("pause");

return EXIT\_SUCCESS;

}

**Output: -**

