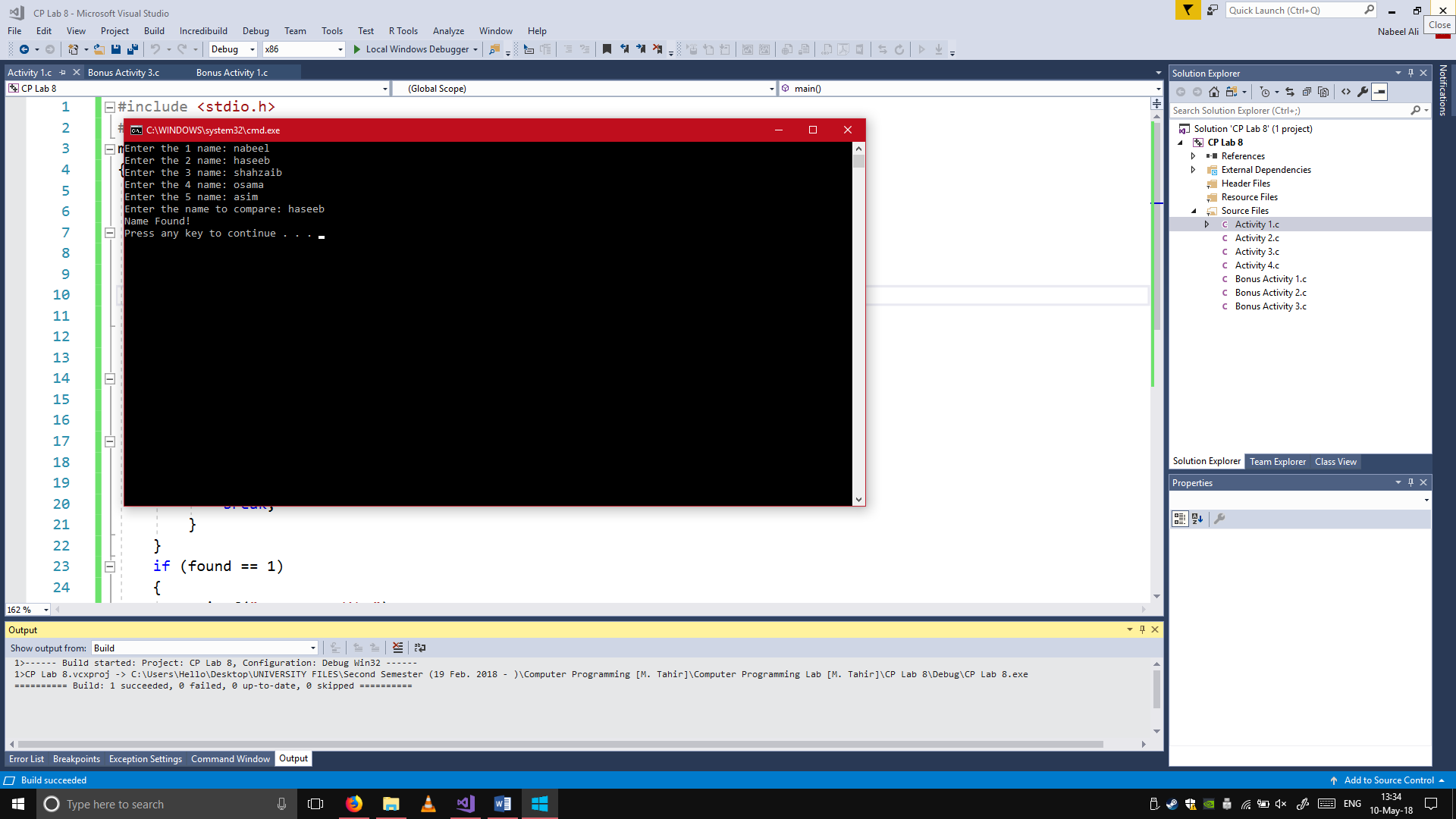
# **Computer Programming Lab 8**

**Nabeel Ali BEE173059 Section-3**

Activity 1:

#include <stdio.h>

#include <string.h>

main()

{

char names[5][40], comparing[40];

int check, found = 0;

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

{

printf("Enter the %d name: ", i+1);

scanf("%s", &names[i]);

}

printf("Enter the name to compare: ");

scanf("%s", comparing);

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

{

check = strcmp(names[k], comparing);

if (check == 0)

{

found = 1;

break;

}

}

if (found == 1)

{

printf("Name Found!\n");

}

else

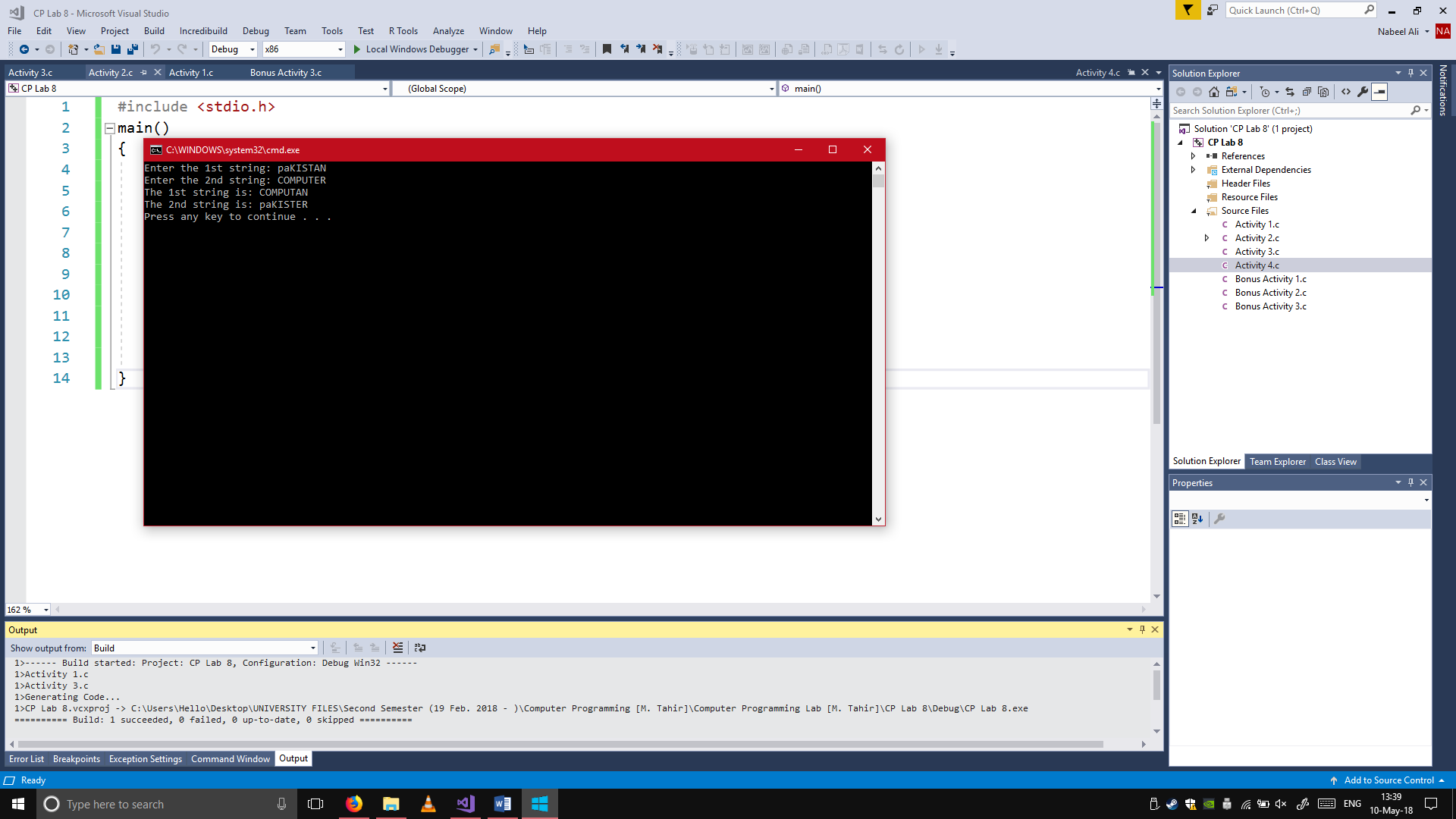
{

printf("Name not found.\n");

}

}

Activity 2:

#include <stdio.h>

main()

{

int X[10], Y[10], Z[10];

printf("Enter the 1st string: ");

scanf("%s", X);

printf("Enter the 2nd string: ");

scanf("%s", Y);

strncpy(Z, X, 5);

strncpy(X, Y, 5);

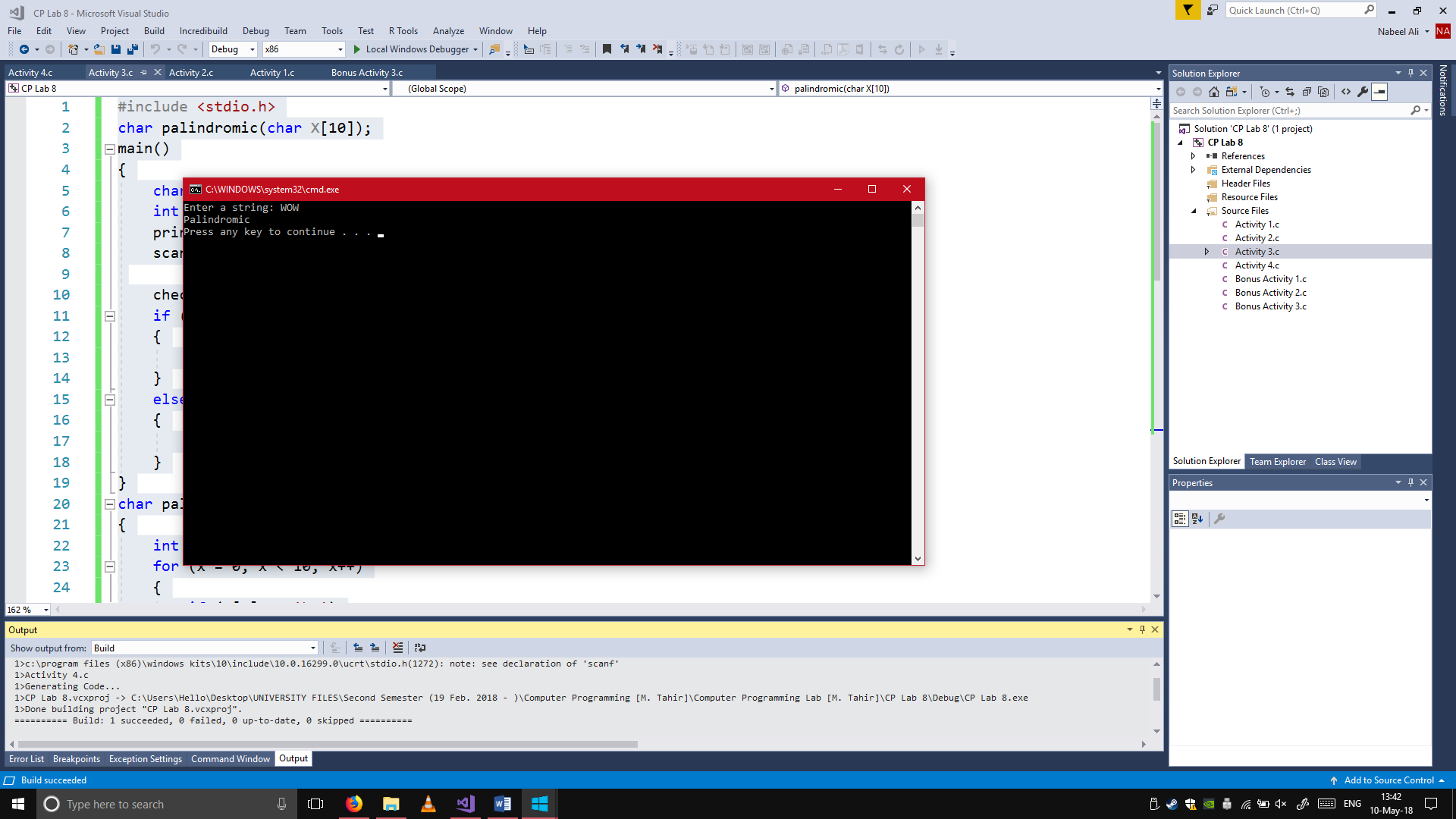
strncpy(Y, Z, 5);

printf("The 1st string is: %s\n", X);

printf("The 2nd string is: %s\n", Y);

}

Activity 3:

#include <stdio.h>

char palindromic(char X[10]);

main()

{

char X[10];

int check;

printf("Enter a string: ");

scanf("%s", &X);

check = palindromic(X);

if (check == 1)

{

printf("Palindromic\n");

}

else

{

printf("NOT palindromic\n");

}

}

char palindromic(char X[10])

{

int x, j, check = 0;

for (x = 0; x < 10; x++)

{

if (X[x] == '\0')

{

j = x - 1;

break;

}

}

for (int i = 0; i < x; i++)

{

if (X[i] != X[j])

{

check = 1;

}

j--;

}

if (check == 0)

{

return 1;

}

else

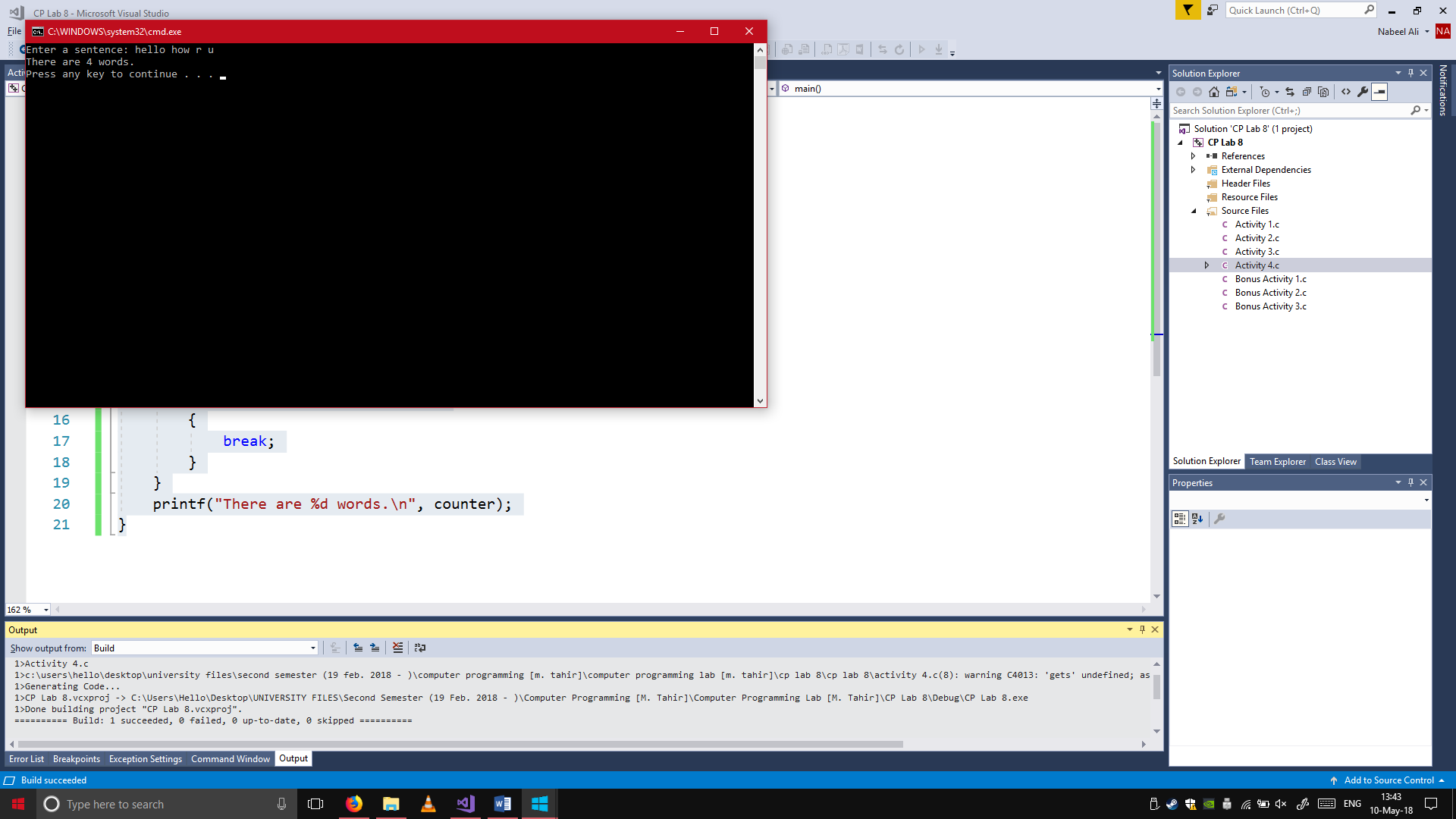
{

return 0;

}

}

Activity 4:

#include <stdio.h>

#include <string.h>

main()

{

char sentence[50];

int counter = 1;

printf("Enter a sentence: ");

gets(sentence);

for (int i = 0; i < 50; i++)

{

if (sentence[i] == ' ')

{

counter++;

}

else if (sentence[i] == '\0')

{

break;

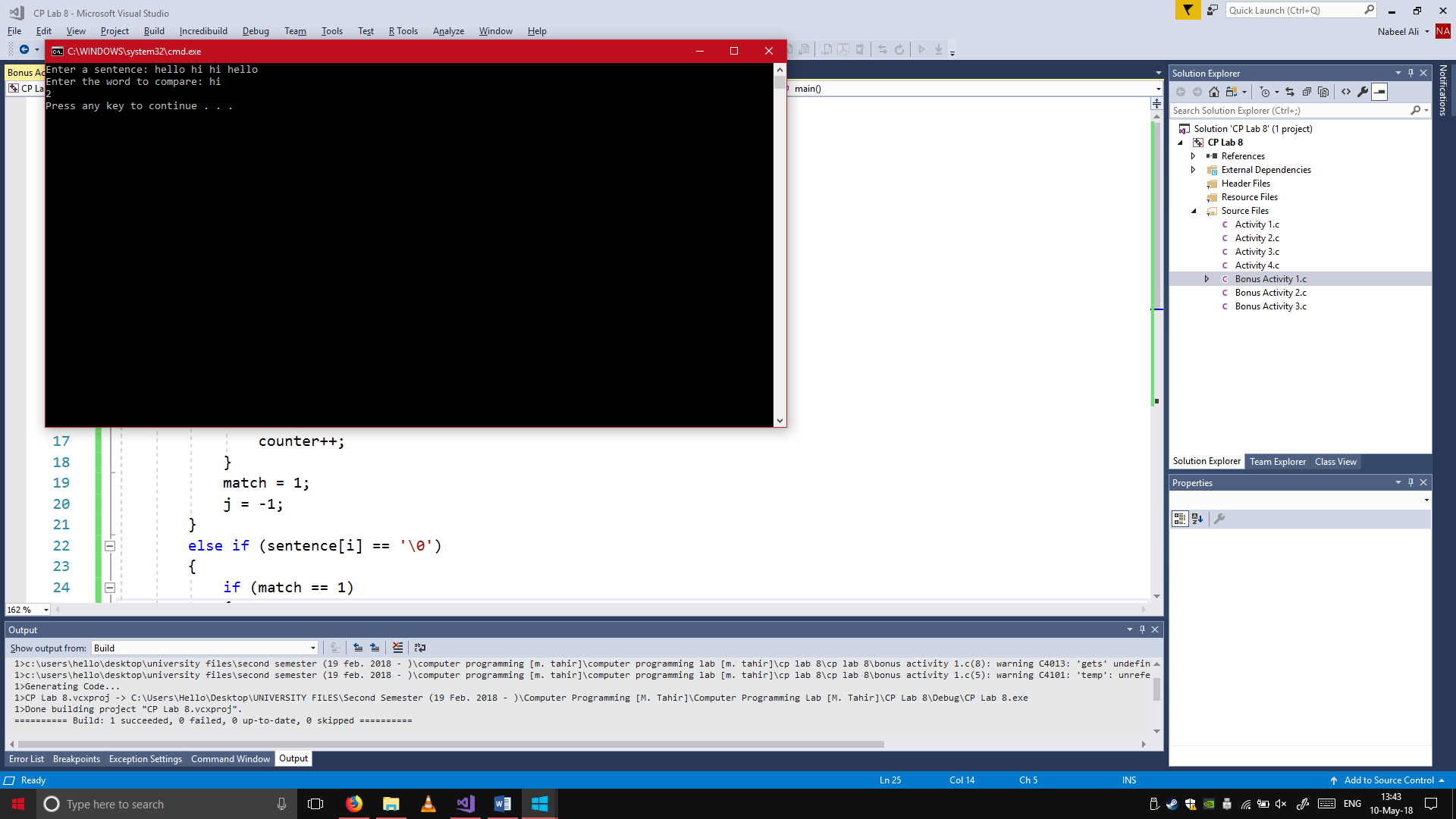
}

}

printf("There are %d words.\n", counter);

}

Bonus Activity 1:

#include <stdio.h>

main()

{

char sentence[50], word[10], temp[10];

int i = 0, j = 0, counter = 0, match = 1;

printf("Enter a sentence: ");

gets(sentence);

printf("Enter the word to compare: ");

gets(word);

for (i = 0; i < 50; i++)

{

if (sentence[i] == ' ')

{

if (match == 1)

{

counter++;

}

match = 1;

j = -1;

}

else if (sentence[i] == '\0')

{

if (match == 1)

{

counter++;

}

break;

}

else if (sentence[i] != word[j])

{

match = 0;

}

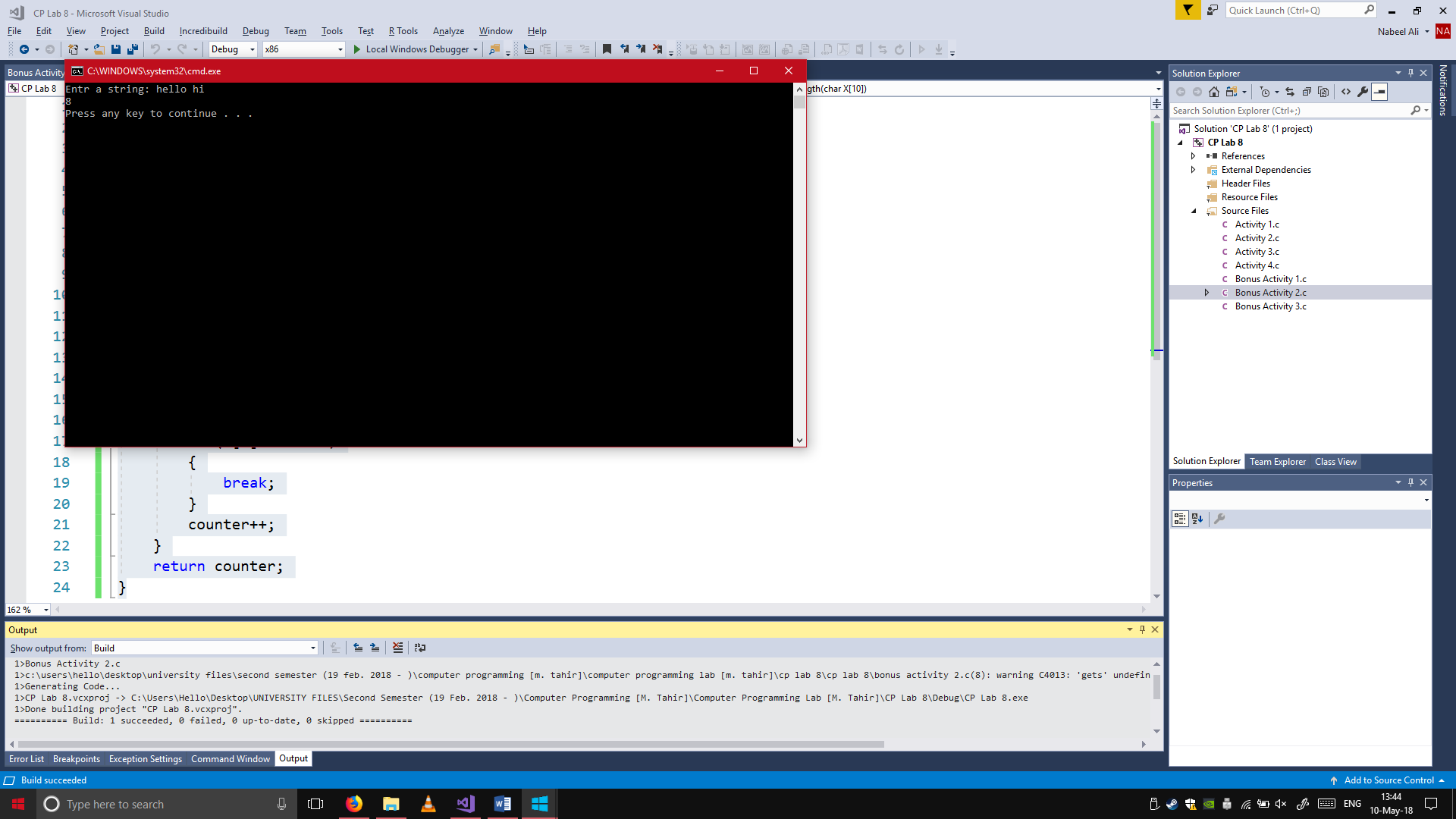
j++;

}

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

}

Bonus Activity 2:

#include <stdio.h>

int length(char X[10]);

main()

{

char X[10];

int l;

printf("Entr a string: ");

gets(X);

l = length(X);

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

}

int length(char X[10])

{

int counter = 0;

for (int i = 0; i < 10; i++)

{

if (X[i] == '\0')

{

break;

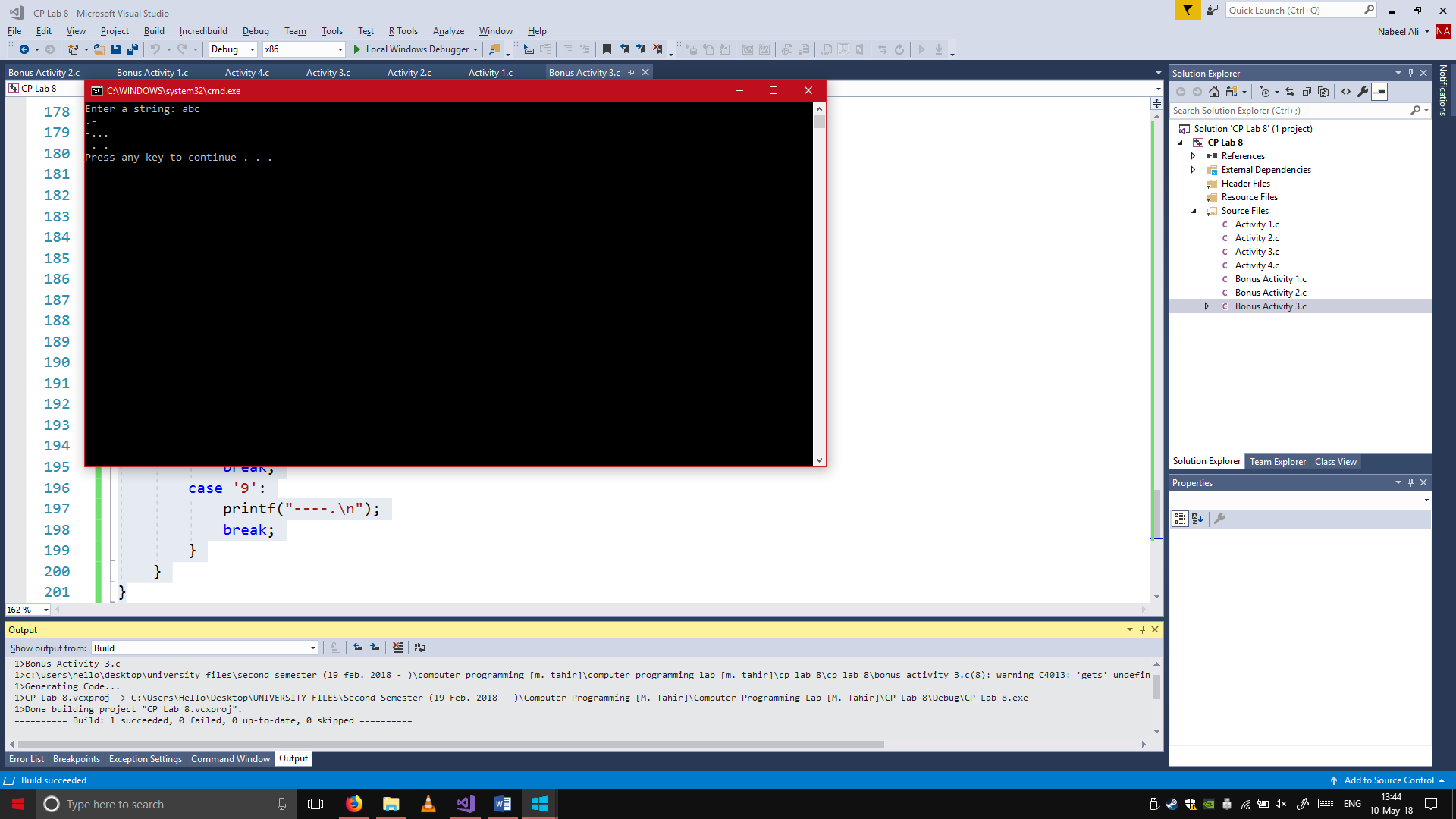
}

counter++;

}

return counter;

Bonus Activity 3:

#include <stdio.h>

main()

{

char X[50];

int num = 0;

printf("Enter a string: ");

gets(X);

for (int i = 0; i < 50; i++)

{

switch (X[i])

{

case 'A':

printf(".-\n");

break;

case 'B':

printf("-...\n");

break;

case 'C':

printf("-.-.\n");

break;

case 'D':

printf("-..\n");

break;

case 'E':

printf(".\n");

break;

case 'F':

printf("..-.\n");

break;

case 'G':

printf("--.\n");

break;

case 'H':

printf("....\n");

break;

case 'I':

printf("..\n");

break;

case 'J':

printf(".---\n");

break;

case 'K':

printf("-.-\n");

break;

case 'L':

printf(".-..\n");

break;

case 'M':

printf("--\n");

break;

case 'N':

printf("-.\n");

break;

case 'O':

printf("---\n");

break;

case 'P':

printf(".--.\n");

break;

case 'Q':

printf("--.-\n");

break;

case 'R':

printf(".-.\n");

break;

case 'S':

printf("...\n");

break;

case 'T':

printf("-\n");

break;

case 'U':

printf("..-\n");

break;

case 'V':

printf("...-\n");

break;

case 'W':

printf(".--\n");

break;

case 'X':

printf("-..-\n");

break;

case 'Y':

printf("-.--\n");

break;

case 'Z':

printf("--..\n");

break;

case 'a':

printf(".-\n");

break;

case 'b':

printf("-...\n");

break;

case 'c':

printf("-.-.\n");

break;

case 'd':

printf("-..\n");

break;

case 'e':

printf(".\n");

break;

case 'f':

printf("..-.\n");

break;

case 'g':

printf("--.\n");

break;

case 'h':

printf("....\n");

break;

case 'i':

printf("..\n");

break;

case 'j':

printf(".---\n");

break;

case 'k':

printf("-.-\n");

break;

case 'l':

printf(".-..\n");

break;

case 'm':

printf("--\n");

break;

case 'n':

printf("-.\n");

break;

case 'o':

printf("---\n");

break;

case 'p':

printf(".--.\n");

break;

case 'q':

printf("--.-\n");

break;

case 'r':

printf(".-.\n");

break;

case 's':

printf("...\n");

break;

case 't':

printf("-\n");

break;

case 'u':

printf("..-\n");

break;

case 'v':

printf("...-\n");

break;

case 'w':

printf(".--\n");

break;

case 'x':

printf("-..-\n");

break;

case 'y':

printf("-.--\n");

break;

case 'z':

printf("--..\n");

break;

case '0':

printf("-----\n");

break;

case '1':

printf(".----\n");

break;

case '2':

printf("..---\n");

break;

case '3':

printf("...--\n");

break;

case '4':

printf("....-\n");

break;

case '5':

printf(".....\n");

break;

case '6':

printf("-....\n");

break;

case '7':

printf("--...\n");

break;

case '8':

printf("---..\n");

break;

case '9':

printf("----.\n");

break;

}

}

}