Saad Teeti

Step 1: Problem Identification and Statement

Develop a software solution that provides the wind chill temperature for air temperature and wind speed entered by the user. This need to be devolved through a set of data entered into the program by the developer and the user should input numbers and should get his windchill temperature.

Step 2: Gathering of Information and Input/Output Description

The Wind-chill Factor is the overall heat transfer coefficient of the small plastic cylinder that Simple and Passel used in their experiments, from the water/ice core at freezing point, to the environment. This number is a function of wind speed only. It had to be multiplied by a temperature difference to get the heat flow, which was called the Wind-chill Index, a three- or four-digit number. To calculate the heat loss rate of the plastic bottle, the temperature difference should be the difference between the freezing point and air temperature.

The data from the table below is given to the developer to design the program with it. The developer must use 2d arrays to store the values in the table. The air temperature is the column part while the Windspeed is the row part. To Calculate the windchill, we subtract the drop point from the user input temperature.

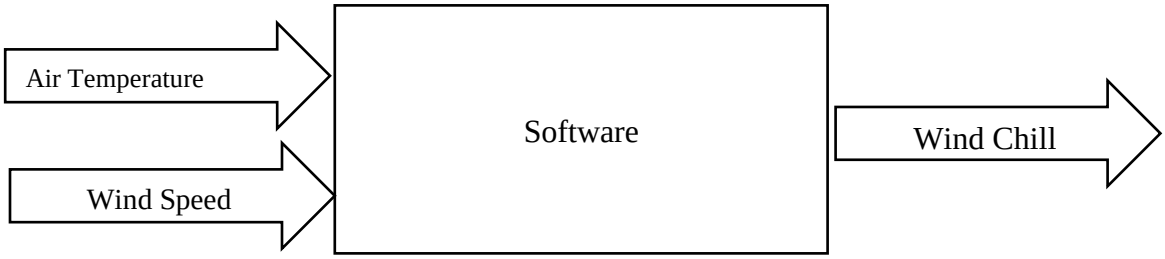
Wind-chill Table

Wind Speed (km/hour) \ Air Temperature (Celsius)	0	-1	-2	-3	-4	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60
6	2.02	2.17	2.32	2.47	2.62	2.77	3.52	4.27	5.02	5.77	6.51	7.26	8.01	8.76	9.51	10.25	11.00
8	2.74	2.91	3.09	3.26	3.44	3.61	4.48	5.36	6.23	7.10	7.97	8.85	9.72	10.59	11.46	12.34	13.21
10	3.31	3.51	3.70	3.90	4.09	4.29	5.26	6.23	7.21	8.18	9.15	10.13	11.10	12.07	13.05	14.02	14.99
15	4.42	4.65	4.88	5.12	5.35	5.58	6.75	7.91	9.08	10.24	11.41	12.57	13.74	14.90	16.07	17.23	18.40
20	5.24	5.50	5.77	6.03	6.29	6.55	7.86	9.17	10.48	11.79	13.10	14.41	15.72	17.02	18.33	19.64	20.95
25	5.91	6.19	6.48	6.76	7.05	7.34	8.76	10.19	11.61	13.04	14.46	15.89	17.31	18.74	20.17	21.59	23.02
30	6.47	6.78	7.08	7.39	7.69	8.00	9.52	11.04	12.57	14.09	15.62	17.14	18.66	20.19	21.71	23.23	24.76
35	6.96	7.28	7.61	7.93	8.25	8.57	10.18	11.79	13.40	15.01	16.62	18.23	19.83	21.44	23.05	24.66	26.27
40	7.40	7.73	8.07	8.41	8.74	9.08	10.77	12.45	14.13	15.82	17.50	19.19	20.87	22.56	24.24	25.93	27.61
45	7.79	8.14	8.49	8.84	9.19	9.54	11.29	13.04	14.80	16.55	18.30	20.06	21.81	23.56	25.31	27.07	28.82
50	8.14	8.50	8.87	9.23	9.59	9.96	11.77	13.59	15.40	17.22	19.03	20.84	22.66	24.47	26.29	28.10	29.92
55	8.47	8.84	9.22	9.59	9.97	10.34	12.21	14.08	15.96	17.83	19.70	21.57	23.44	25.31	27.19	29.06	30.93
60	8.77	9.16	9.54	9.93	10.31	10.70	12.62	14.54	16.47	18.39	20.32	22.24	24.17	26.09	28.02	29.94	31.86
65	9.05	9.45	9.84	10.24	10.63	11.03	13.00	14.97	16.95	18.92	20.90	22.87	24.84	26.82	28.79	30.76	32.74
70	9.32	9.72	10.13	10.53	10.93	11.34	13.36	15.38	17.40	19.42	21.44	23.46	25.48	27.50	29.52	31.54	33.55
75	9.57	9.98	10.39	10.80	11.22	11.63	13.69	15.76	17.82	19.88	21.95	24.01	26.07	28.14	30.20	32.26	34.33
80	9.80	10.22	10.64	11.06	11.49	11.91	14.01	16.11	18.22	20.32	22.43	24.53	26.64	28.74	30.84	32.95	35.05
85	10.03	10.45	10.88	11.31	11.74	12.17	14.31	16.46	18.60	20.74	22.88	25.03	27.17	29.31	31.46	33.60	35.74
90	10.24	10.67	11.11	11.55	11.98	12.42	14.60	16.78	18.96	21.14	23.32	25.50	27.68	29.86	32.04	34.22	36.40
95	10.44	10.88	11.33	11.77	12.21	12.66	14.87	17.09	19.30	21.52	23.74	25.95	28.17	30.38	32.60	34.81	37.03
100	10.64	11.09	11.54	11.99	12.43	12.88	15.13	17.38	19.63	21.88	24.13	26.38	28.63	30.88	33.13	35.38	37.63

The above table provides the drop in temperature for given wind speeds and air temperatures.

Input/output Description:

The program has two inputs and one output. The two inputs are temperature of the air(In Celsius) and the windspeed (In Km/H). The program will output the Wind Chill variable. This is explained in the following diagram:



Test Cases and Algorithm:

Test Case 1: Testing the maximum and the minimum values for air temperature and wind speed

To exit the program press 1
Please enter the temperature in Celsius
0
Please enter the wind speed in km/h
6 the Wind Chill is -2.02
To exit the program press 1
Please enter the temperature in Celsius

Test Case 2: Testing the maximum and minimum values for air temperature and wind speed

To exit the program press 1
Please enter the temperature in Celsius
-60
Please enter the wind speed in km/h
100
the Wind Chill is -97.63 To
exit the program press 1
Please enter the temperature in Celsius

Test Case 3: Testing values equal to those in the wind-chill table headers for air-temperature and wind speed

To exit the program press 1
Please enter the temperature in Celsius
-2
Please enter the wind speed in km/h
10
the Wind Chill is -5.7

To exit the program press 1
Please enter the temperature in Celsius

Test Case 4: Testing values equal to those in the wind-chill table headers for air-temperature and wind speed
To exit the program press 1
Please enter the temperature in Celsius
-5
Please enter the wind speed in km/h
25 the Wind Chill is -12.34
To exit the program press 1
Please enter the temperature in Celsius

Test Case 5: Testing mid-point values for air temperature and wind speed. A mid point value is one exactly between two values provided in the table.
To exit the program press 1
Please enter the temperature in Celsius
-1.5
Please enter the wind speed in km/h
9 the Wind Chill is -5.01
To exit the program press 1
Please enter the temperature in Celsius

Test Case 6: Testing values below the mid-points for air temperature and wind speed.
To exit the program press 1
Please enter the temperature in Celsius
-1.8
Please enter the wind speed in km/h
21
the Wind Chill is -7.57 To
exit the program press 1
Please enter the temperature in Celsius
Test Case 7: Testing values above the mid-points for air temperature and wind speed.

To exit the program press 1
Please enter the temperature in Celsius
-1.2
Please enter the wind speed in km/h

24 the Wind Chill is -7.39
To exit the program press 1
Please enter the temperature in Celsius
Test Case 8: Test invalid values for the air temperature (outside first extreme)
To exit the program press 1
Please enter the temperature in Celsius
5
Tempreture is out of the range:
Please input tempreture between: 0 and -60
Test Case 9: Test invalid values for the windspeed
To exit the program press 1
Please enter the temperature in Celsius
0
Please enter the wind speed in km/h
120
Windspeed is out of the range
Please input windspeed between: 6 and 100
Test Case 10: Test invalid values for the air temperature (outside second extreme).

To exit the program press 1
Please enter the temperature in Celsius
-70
Tempreture is out of the range:
Please input tempreture between: 0 and -60
Test Case 11: Windspeed is lower than 6 and outside the range
To exit the program press 1
Please enter the temperature in Celsius
-5
Please enter the wind speed in km/h
-1
Windspeed is out of the range
Please input windspeed between: 6 and 100

Test error conditions that can occur in the wind-chill file:

Test Case 12: No chill.txt file present. *Error opening the file*
Press any key to continue . . .
Test Case 13: Number of values in a temperature drop line not equal to the number of values in the first line too small.
Error Loading the Matrix 101Press any key to continue . . .

Test Case 14 • Number of lines containing temperature drop values is not equal to the number of
values in the second line (too many and too few). *Eror*
Loading the Matrix
Press any key to continue . . .

Test Case 15: Number of values in first line exceeds the size of the array used to store the values.
The file will crash and not open.
Test Case 16: Number of values in the second line exceeds the size of the array used to store the values.
The file will crash clash and not open
Test Case 17: Number of values in a temperature drop line exceeds the column dimension of the matrix used to store the temperature drop values.
The file will crash will clash and not open
Test Case 18 *:* ***Number of lines containing the temperature drop values exceeds row dimension of the matrix used to store the temperature drop values.*** *The file will crash and will not open.*

Test Case 19: Trying the code on the data in chillsmall file.
To exit the program press 1
Please enter the temperature in Celsius
-7
Please enter the wind speed in km/h
20 the Wind Chill is -13.55
To exit the program press 1
Please enter the temperature in Celsius

Test Case 20: Trying the code on the data in chillsmall file and trying minimum and maximum values.

To exit the program press 1
Please enter the temperature in Celsius
-4
Please enter the wind speed in km/h
15 the Wind Chill is -9.35
To exit the program press 1
Please enter the temperature in Celsius

Test Case 21: Trying the code on the data in chillsmall file and trying minimum and maximum values.

To exit the program press 1

Please enter the temperature in Celsius
-35
Please enter the wind speed in km/h
50 the Wind Chill is -55.84
To exit the program press 1
Please enter the temperature in Celsius

Test Case 22: Trying the code on the data in chillsmall file and putting out of range tempreture(the first extreme).

To exit the program press 1
Please enter the temperature in Celsius
-3
Tempreture is out of the range:
Please input tempreture between: -4 and -35

Test Case 23: Trying the code on the data in chillsmall file and putting out of range windspeed.

To exit the program press 1
Please enter the temperature in Celsius
-10
Please enter the wind speed in km/h
51
Windspeed is out of the range
Please input windspeed between: 15 and 50

Test Case 24: Trying the code on the data in chillsmall file and putting out of range tempreture(the second extreme).

To exit the program press 1
Please enter the temperature in Celsius
-36
Tempreture is out of the range:
Please input tempreture between: -4 and -35

Algorithm

Main program

Declare row and column as integers and assign 0 to each one of them.
Declare rowindex and columnindex as integers
Declare finalvalue as double
Declare array temperature of size 40, array windspeed of size 40, and 2d array windchill of size 40 by 40, as doubles.
Declare usertempreture, userwindspeed as doubles
Declare userinput as a string

Declare exit as a boolian and assign false to it
Call function: LoadData(temperature, windspeed, windchill, row, column).
Open an infinite loop:
Print "To exit the program press 1"
Print "Please enter the temperature in Celsius"
Read the value into usertempreture
If usertempreture is equal to 1, exit the program
Otherwise call function: TemperatureCheck(temperature, column, usertempreture)
Print "Please enter the wind speed in km/h"
Read the value into userwindspeed
If userwinspeed is equal to 1, exit the program
Otherwise call function: WindCheck(windspeed, row, userwindspeed) Assign the return value from the function ColumnNumber(temperature, column, usertempreture) to columnindex
Assign the return value from the function RowNumber(windspeed, row, userwindspeed) to rowindex
Assign the return value from the function WindChill (windchill,columnindex, rowindex, usertempreture) to finalvalue Print "the Wind Chill is" finalvalue Return

LoadData(temperature array, windspeed array, windchill 2d array, row, column).
Declare count as integer and assign 0 to it
Declare line as a string
Create a buffer to open a chill.txt file
Open chill.txt file using the buffer
Check if the files open correctly
If not print an error message "Error opening the file", and exit the program
Otherwise: read the values from the file into temperature[column] increment column
Repeat until reaching new line character read the values from the file into windspeed[row]
increment row
Repeat until reaching new line character
Assign 0 to integer k, for k smaller than row:
    For int i initially equal to 0, I smaller than column:
    Read the value into windchill [k][i]
    Increment i
Increment k
Return pointer to the beginning of the line
Repeat and use function getline to put data from file to line
increment count
If row does equal to count – 2, print error message " Error loading the data", then exit the program.
Return pointer to the beginning of the line

Declare llvalues and Linevalues as integers and assign 0 to both of them.
Repeat until the end of the file:
Get a character from the file, if it is a newline character or a space, increament all values.

Assign (Allvalues - row - column) / (count-2) to Linevalues
if column doesn't equal to Linevalues, print eroor message" Error loading the Matrix 101", and exit the program

TemperatureCheck(temperature array, column, usertempreture):
While usertempreture is greater than the first element of the temperature array or smaller than the columnth element of the temperature array:
Print: "Temperature is out of the range"
Print "Please input temperature between" first element of the temperature array (temperature[0]) "and" columnth element of the temperature array (temperature[column])
Read the value into usertempreture

WindCheck(windspeed array, row, userwindspeed):
While userwindspeed is smaller than the first element of the windspeed array or greater than the (row-1)th element of the windspeed array:
Print: "Wind speed is out of the range"
Print "Please input wind speed between" first element of the windspeed array (temperature[0]) "and" (row-1)th element of the windspeed array (temperature[row-1])
Read the value into userwindspeed

ColumnNumber(temperature array, column, usertempreture)
Declare integer index and assign 0 to it
For int i, smaller than (column-1), which is initially equal to 0
If usertempreture is smaller or equal to the ith element of the temperature array and usertempreture is greater than the (i+1)th element of the temperature array, then:
Declare the average as a double
Assign the average of the ith and the (i+1)th element of the temperature array to the average
If the usertempreture is greater or equal to average, assign i to index
Else if the usertempreture is smaller than the average, assign i+1 to index
Increment i
Return the value of index

RowNumber(windspeed array, row, userwindspeed)
Declare integer index and assign 0 to it
For int i, smaller than (row-1), which is initially equal to 0

If userwindspeed is greater or equal to the i^{th} element of the windspeed array and userwindspeed is smaller than the $(i+1)^{th}$ element of the windspeed array, then:

Declare the average as a double

Assign the average of the i^{th} and the $(i+1)^{th}$ element of the windspeed array to the average

If the userwindspeed is smaller than the average, assign ito index

Else if the userwindspeed is greater or equal to average, assign i+1 to index

Return the value of index

WindChill (windchill array, columnindex, rowindex, usertempreture):

Declare the finalvalue as a double

Assign the difference between the usertempreture and the windchill[rownum][colnum] to the finalvalue

Return the value of the finalvalue

Code

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <string> using
namespace std;
void LoadData(double[], double[], double[40][40], int &row, int &column);
void TemperatureCheck(double[], int column, double usertempreture); void
WindCheck(double[], int row, double userwindspeed); int
ColumnNumber(double[], int column, double usertempreture); int
RowNumber(double[], int row, double userwindspeed);;
double WindChill(double[40][40], int colnum, int rownum, double usertempreture); int
main()
{
	//declaring variables
int row = 0, column = 0;    int
rowindex, columnindex;
double finalvalue;
	//declaring arrays, and variables
 double temperature[40], windspeed[40], windchill[40][40], usertempreture,
userwindspeed;
	string userinput;
	bool exit = false;
	// calling load data function to open the data from the file and put it into
arrays
	LoadData(temperature, windspeed, windchill, row, column);
//Using an infenite loop to keep asking the user for inputs   while
(!exit) {
		//asking the user to input temperature          cout <<
"To exit the program press 1" << endl;          cout << "Please enter
the temperature in Celsius" << endl;
		cin >> usertempreture;
		// if the user enter 1 then the program will exit
		if (usertempreture == 1) {
			exit = true;
		break;
		}
		//calling function to check the temperature range
		TemperatureCheck(temperature, column, usertempreture);
//asking the user for wind speed as an input
		cout << "Please enter the wind speed in km/h" << endl;
		cin >> userwindspeed;
	if (userwindspeed == 1) {
	exit = true;
break;
		}
		// calling functions windcheck
		WindCheck(windspeed, row, userwindspeed);
//assigning function columnNumber to columnindex
		columnindex = ColumnNumber(temperature, column, usertempreture);
		//assigning function RowNumber to rowindex            rowindex =
RowNumber(windspeed, row, userwindspeed);              //assigning Windchill function
to final value            finalvalue = WindChill(windchill, columnindex, rowindex,
usertempreture);          cout << " the Wind Chill is " << finalvalue << endl;
```

```
	}



	system("pause");
return 0;
}
// opening the function to load data
void LoadData(double temperature[], double windspeed[], double windchill[40][40], int
&row, int &column)
{

	int count = 0;
string line;		//
opening file
ifstream infile;
	//checking if the files open or not
infile.open("chill.txt", ios::in);
	if (infile.fail())
	{
		//print message if the file is not opening
		cerr << "Error opening the file" << endl;
	system("pause");		exit(0);
	}
	// do loop to store data from file into array

	do
	{

		infile >> temperature[column];
		column++;
	} while (infile.get() != '\n');

	// do loop to store data from file into array
	do
	{
		infile >> windspeed[row];
		row++;
	} while (infile.get() != '\n');
	//for loop to store data from file into 2d array

	for (int k = 0; k < row; k++)
		for (int i = 0; i < column; i++)
		{
			infile >> windchill[k][i];
		}
	// function used to return the pointer to the begining of the
line	infile.clear();	infile.seekg(0, ios::beg);
	// functtion to read data from file into a line
while (getline(infile, line))
		count++;
	// we use this to find Number of lines containing temperature drop values is not
equal to the number of values in the second line.
```

```
	if (row != count - 2) {
		//print error message		cerr <<
" Eror Loading the Matrix" << endl;
system("pause");
		exit(-1);
	}
	// function used to return the pointer to the begining of the
line	infile.clear();	infile.seekg(0, ios::beg);	int
Allvalues = 0, Linevalues = 0;
	// We use this to see if Number of values in a temperature drop line not equal to
the number of values in the first line too small.
	while (!infile.eof())
	{
		infile.get();
		if (infile.peek() == '\n' || infile.peek() == ' ')
			Allvalues++;
	}

	Linevalues = (Allvalues - row - column) / (count - 2);
	if (column != Linevalues) {
//print error message
		cerr << " Error Loading the Matrix 101";
system("pause");
		exit(-1);
	}
}



void TemperatureCheck(double temperature[], int column, double usertempreture) {
	//this function we use to check if the temperature is in the range
	while ((usertempreture > temperature[0]) || (usertempreture < temperature[column -
1]))
	{
		// if not, the following would be printed on the screen.

		cout << "Tempreture is out of the range:" << endl;
		cout << "Please input tempreture between:  " << temperature[0] << " and "
<< temperature[column - 1] << endl;
		cin >> usertempreture;
	}
}
void WindCheck(double windspeed[], int row, double userwindspeed) {
//this function is to check if the user windspeed is in the range
	while ((userwindspeed < windspeed[0]) || (userwindspeed > windspeed[row - 1]))
	{
		// if not, the following would be printed on the screen.
		cout << "Windspeed is out of the range" << endl;		cout <<
"Please input windspeed between:  " << windspeed[0] << " and " << windspeed[row - 1]
<< endl;
		cin >> userwindspeed;
	}
} int ColumnNumber(double temperature[], int column, double
usertempreture) {
```

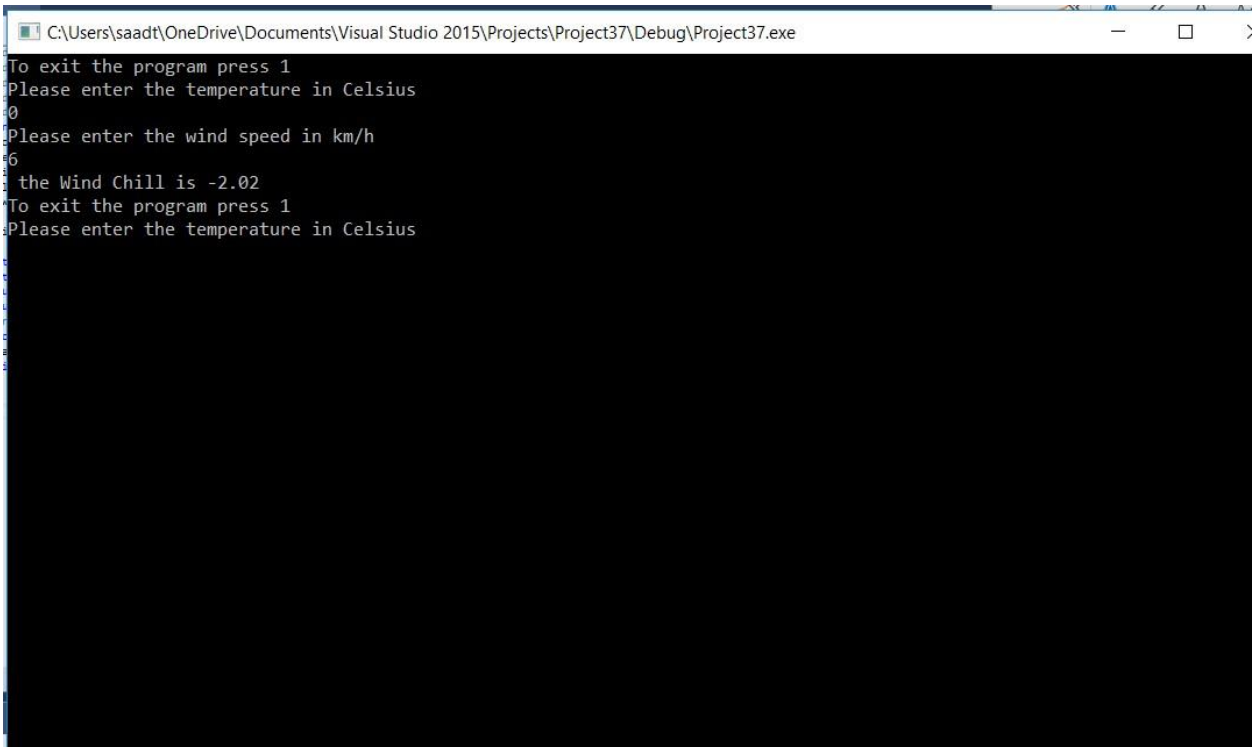
```
	// we use this function to find the closest number for the number that the user enter for temperature if it is not on the chart
	//the equation is to round the  number to the number next to it or before it on the chart, depending on the number, if it it is a midpoint then it will round up, if it is smaller than midpoint it will round down and if it is greater than midpoint it will round up
	int index = 0;
	for (int i = 0; i < column - 1; i++)
	{
		if ((usertempreture <= temperature[i]) && (usertempreture >= temperature[i + 1]))
		{
			double average;
			average = (temperature[i] + temperature[i + 1]) / 2;
			if (usertempreture >= average)
			{
				index = i;
			}
			else if ((usertempreture < average))
			{
				index = i + 1;
			}
		}
	}
	return(index);
}
int RowNumber(double windspeed[], int row, double userwindspeed) {
	{
		// we use this function to find the closest number for the number that the user enter for windspeed if it is not on the chart
		// the equation is to round the  number to the number next to it or before it on the chart, depending on the number, if it it is a midpoint then it will round up, if it is smaller than midpoint it will round down and if it is greater than midpoint it will round up


		int index = 0;
		for (int i = 0; i <= row - 1; i++)
		{
			 if ((userwindspeed >= windspeed[i]) && (userwindspeed <= windspeed[i + 1]))
			{
				double average;
				average = (windspeed[i] + windspeed[i + 1]) / 2;
				if (userwindspeed < average)
				{
					index = i;
				}
				else if ((userwindspeed >= average))
				{
					index = i + 1;
				}
			}
		}
```

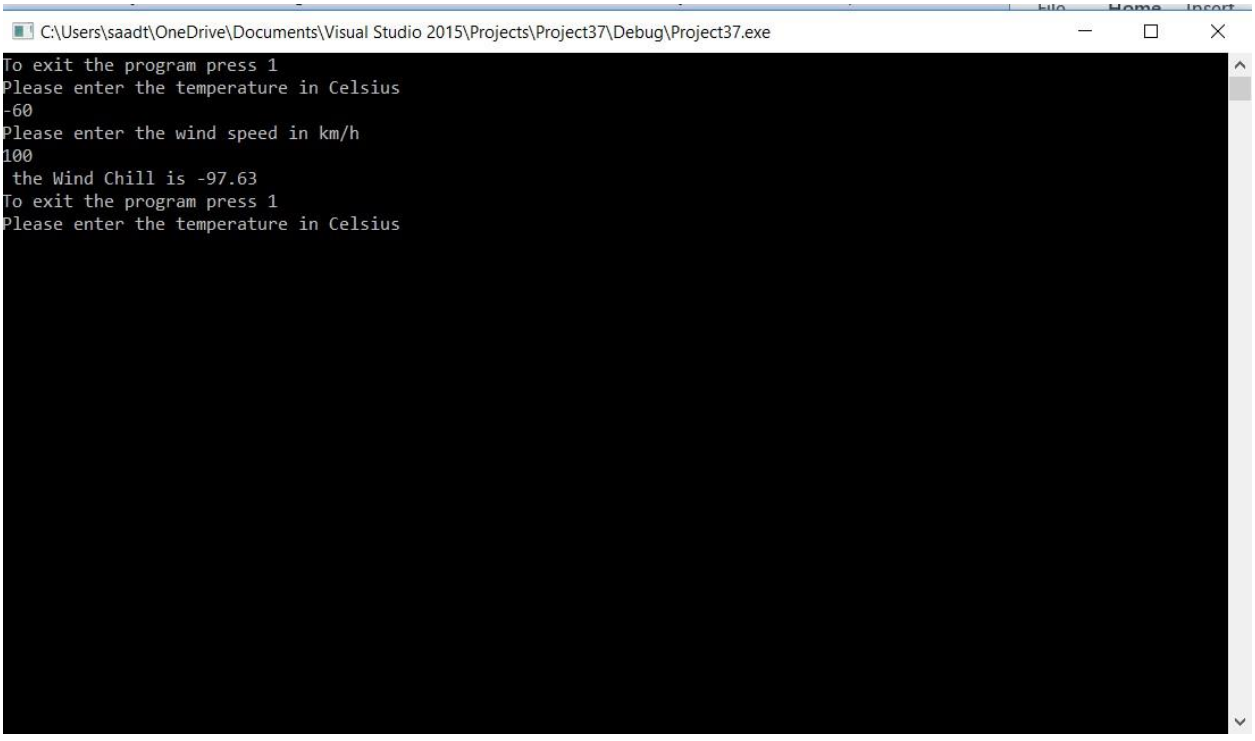
```
                return(index);
        }
}
double WindChill(double windchill[40][40], int colnum, int rownum, double usertempreture)
{
        // we subtract the returned array whcih is windchill[rownum][colnum] from
the temperature that user inputed and  will return it into final value      double
finalvalue;
        finalvalue = usertempreture - windchill[rownum][colnum];
        return finalvalue;
}
```

Test Case

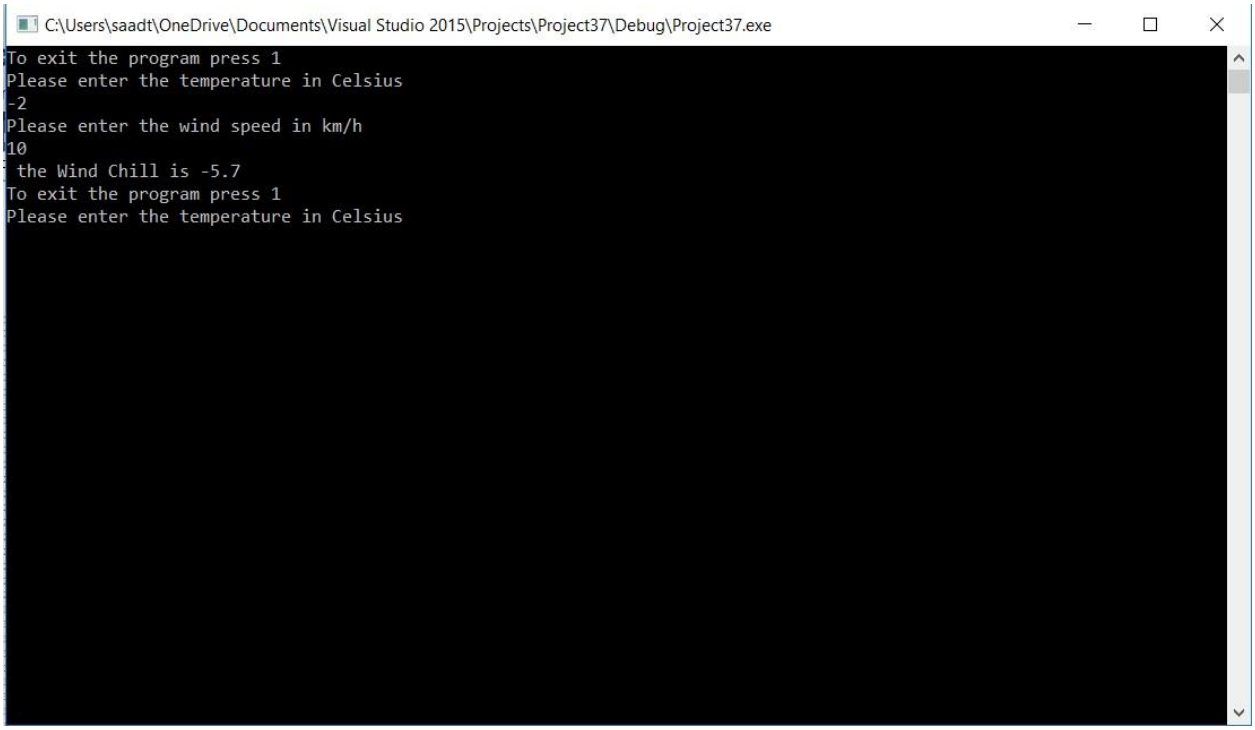
Test Case 1 :



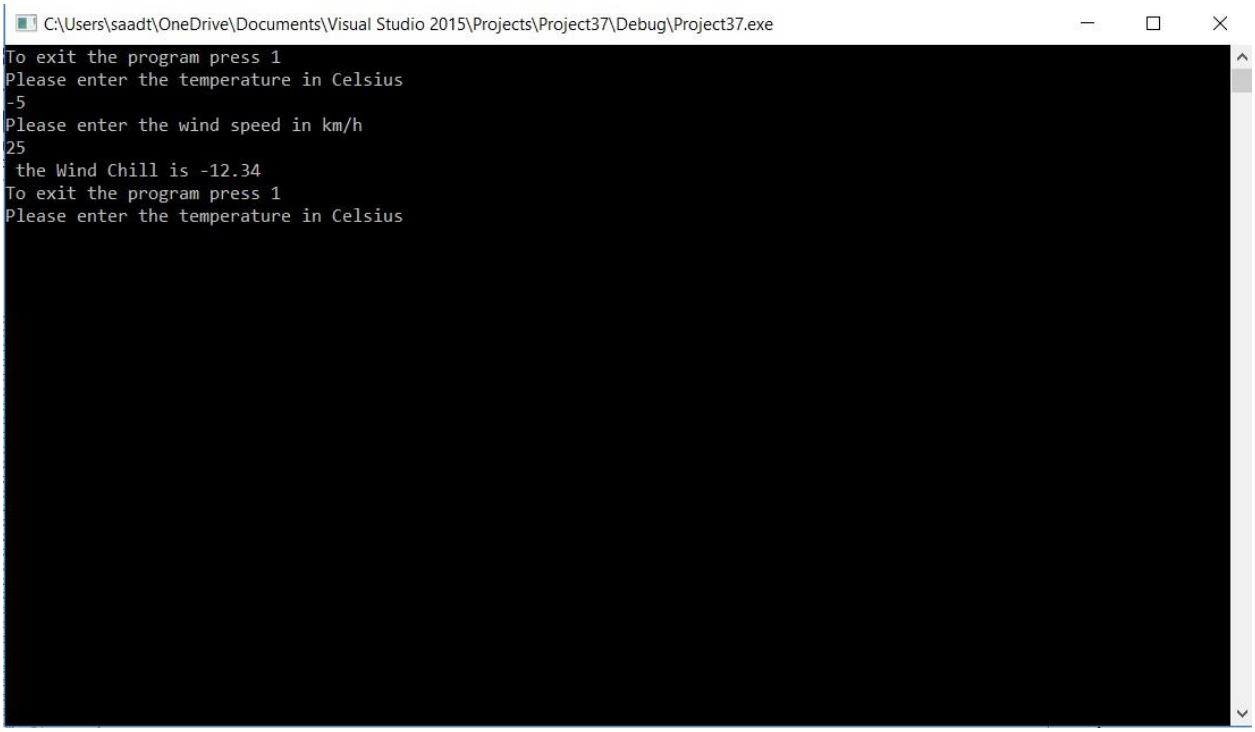
Test case 2:



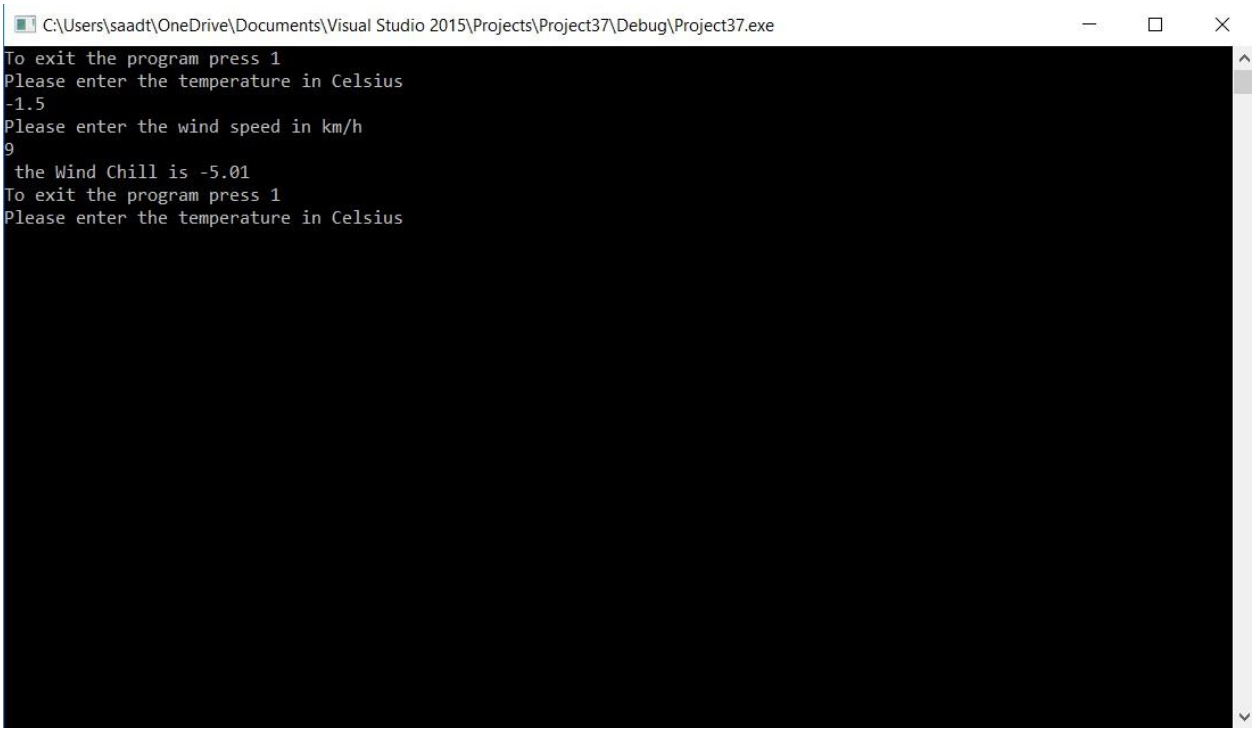
Test Case: 3



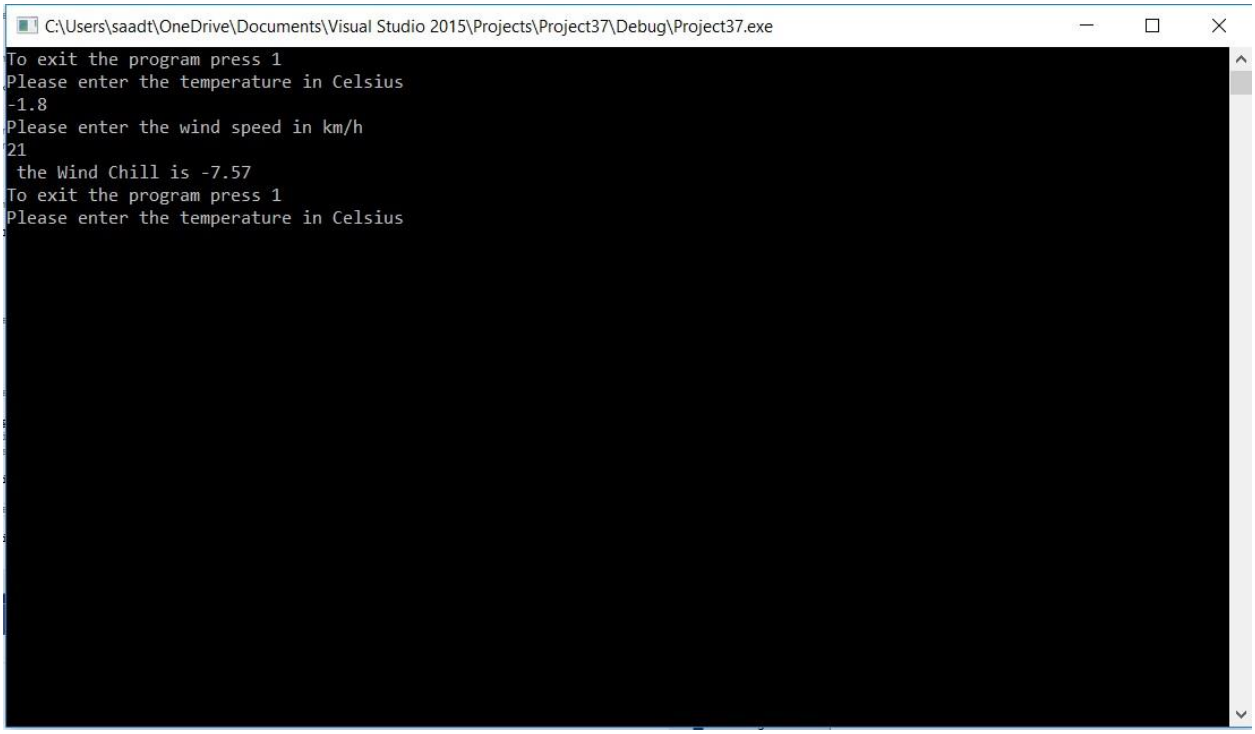
Test Case: 4



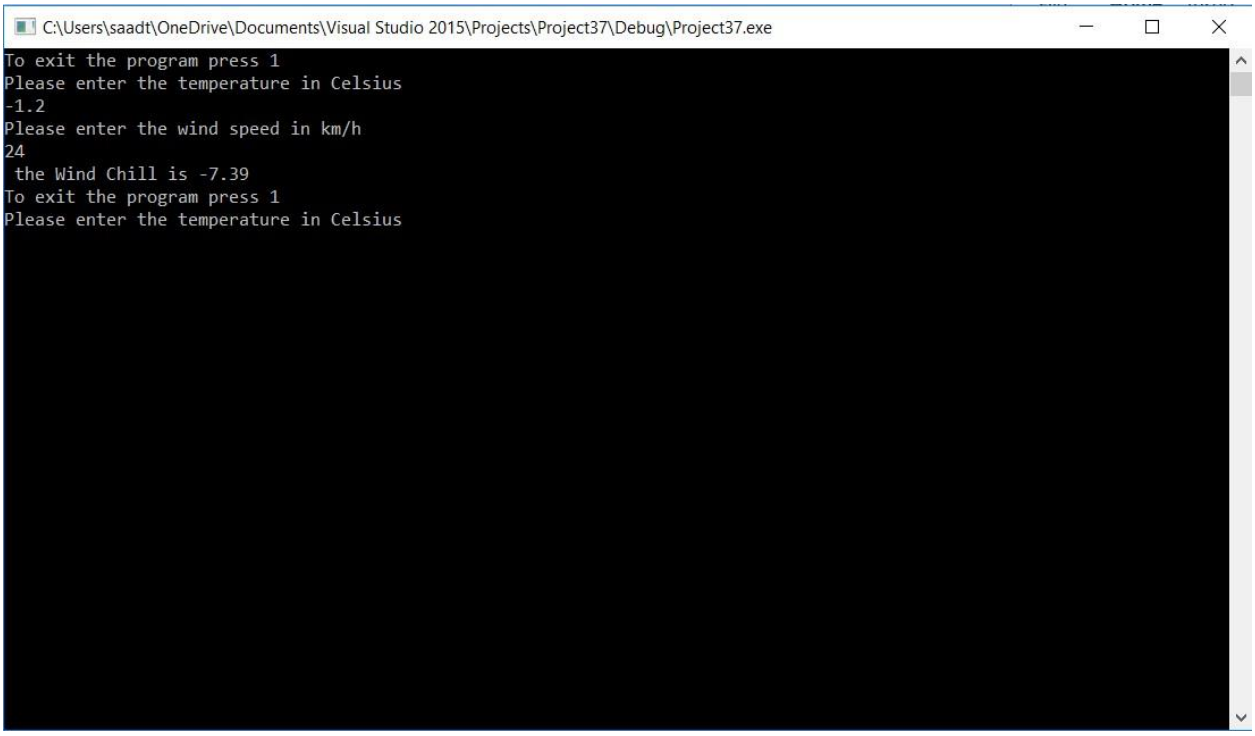
Test Case:5



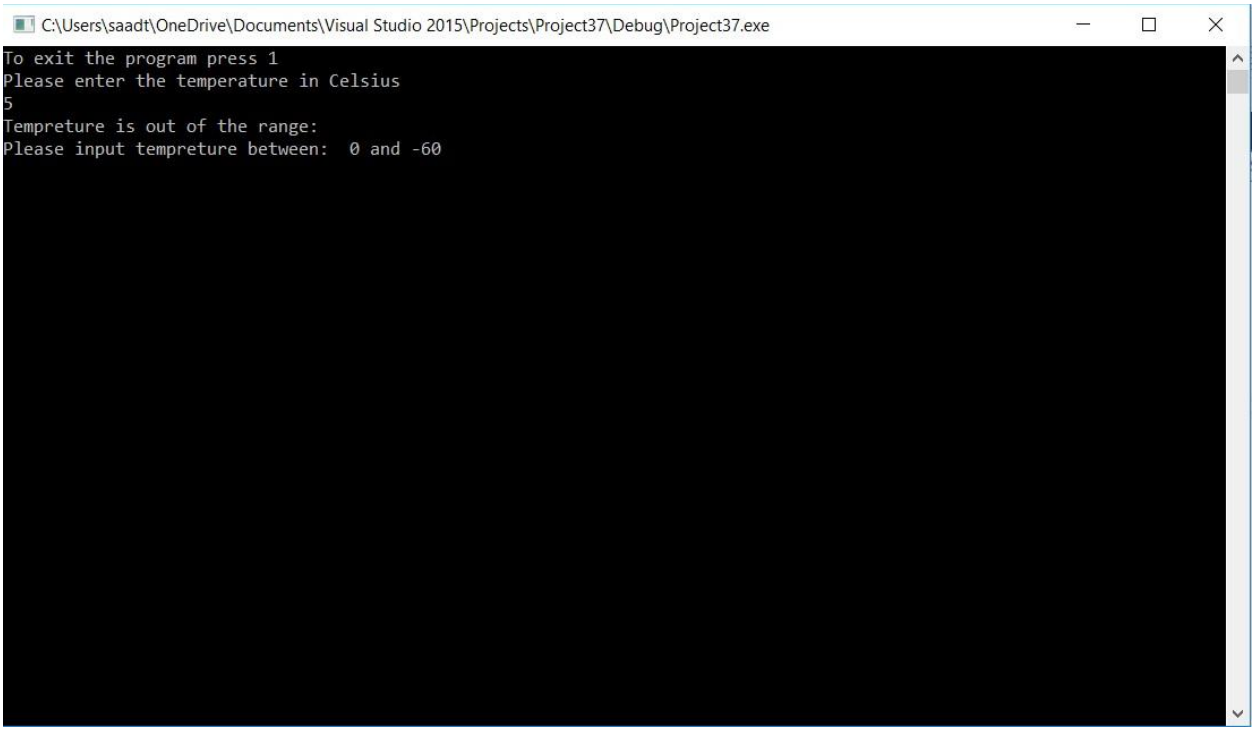
Test Case 6



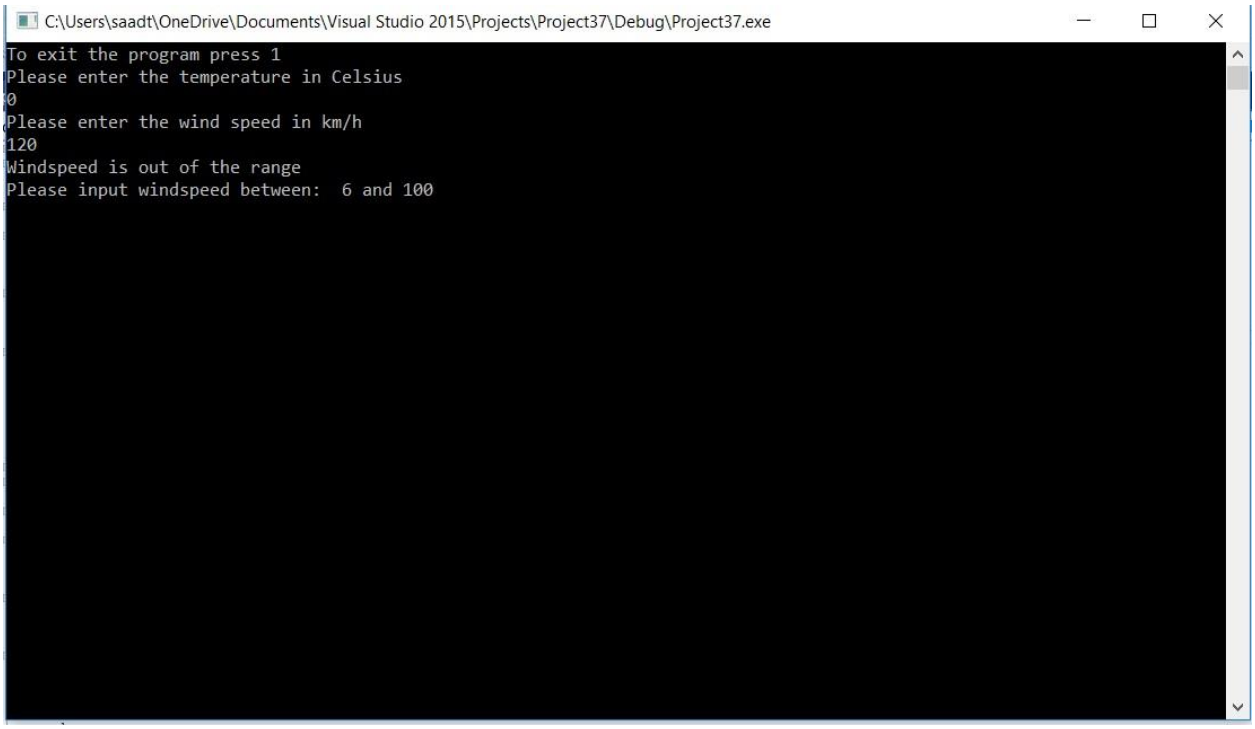
Test Case 7



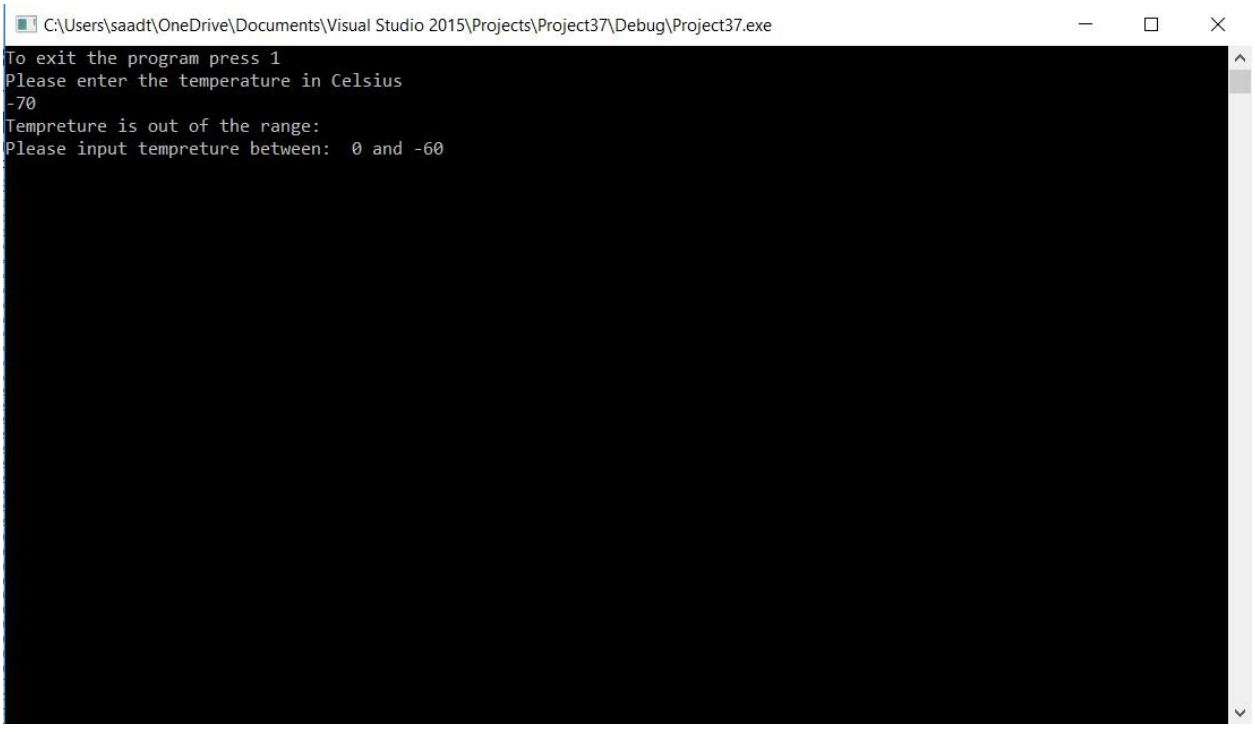
Test Case 8



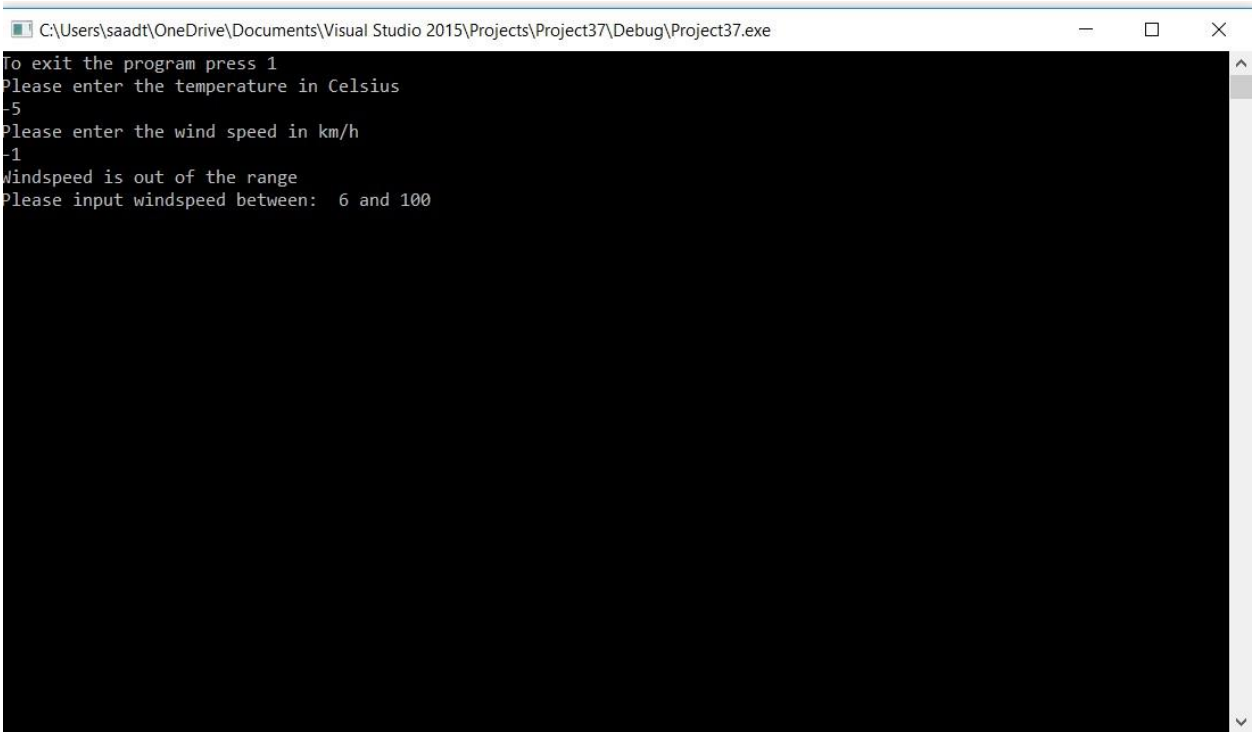
Test Case 9


C:\Users\saadt\OneDrive\Documents\Visual Studio 2015\Projects\Project37\Debug\Project37.exe
To exit the program press 1
Please enter the temperature in Celsius
0
Please enter the wind speed in km/h
120
Windspeed is out of the range
Please input windspeed between: 6 and 100

Test Case 10

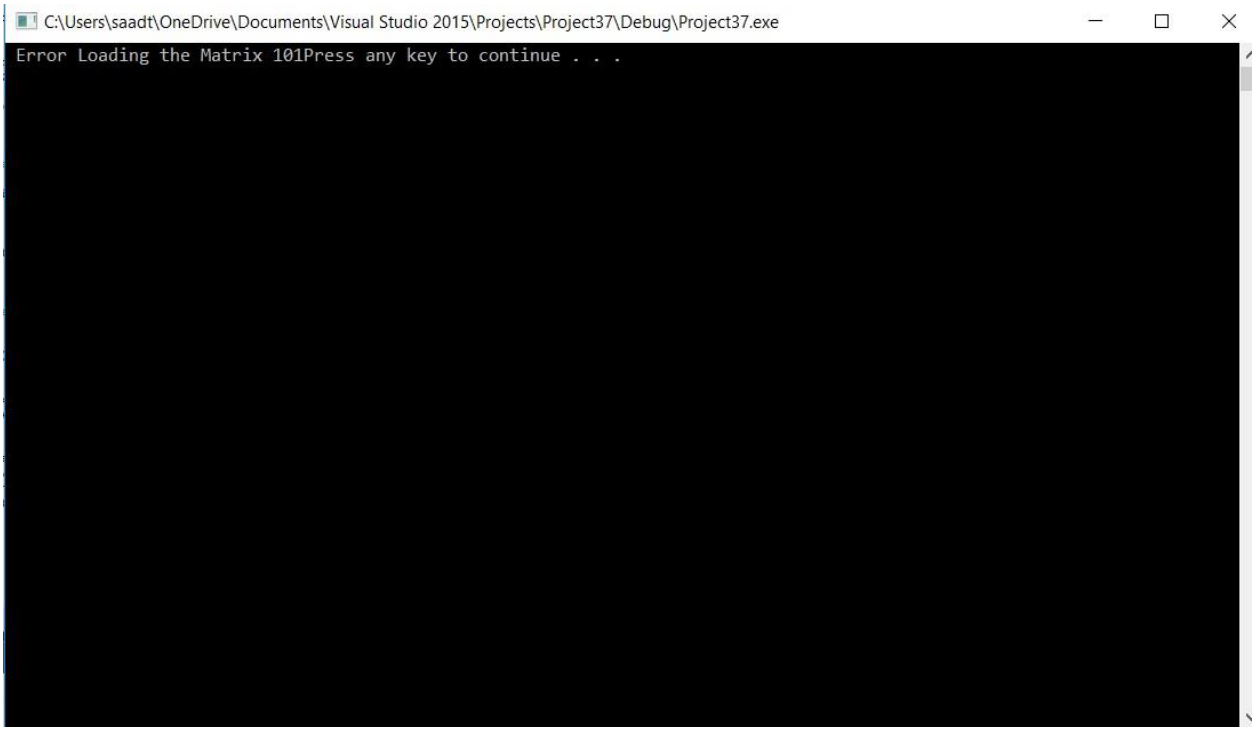

C:\Users\saadt\OneDrive\Documents\Visual Studio 2015\Projects\Project37\Debug\Project37.exe
To exit the program press 1
Please enter the temperature in Celsius
-70
Tempreture is out of the range:
Please input tempreture between: 0 and -60

Test Case 11

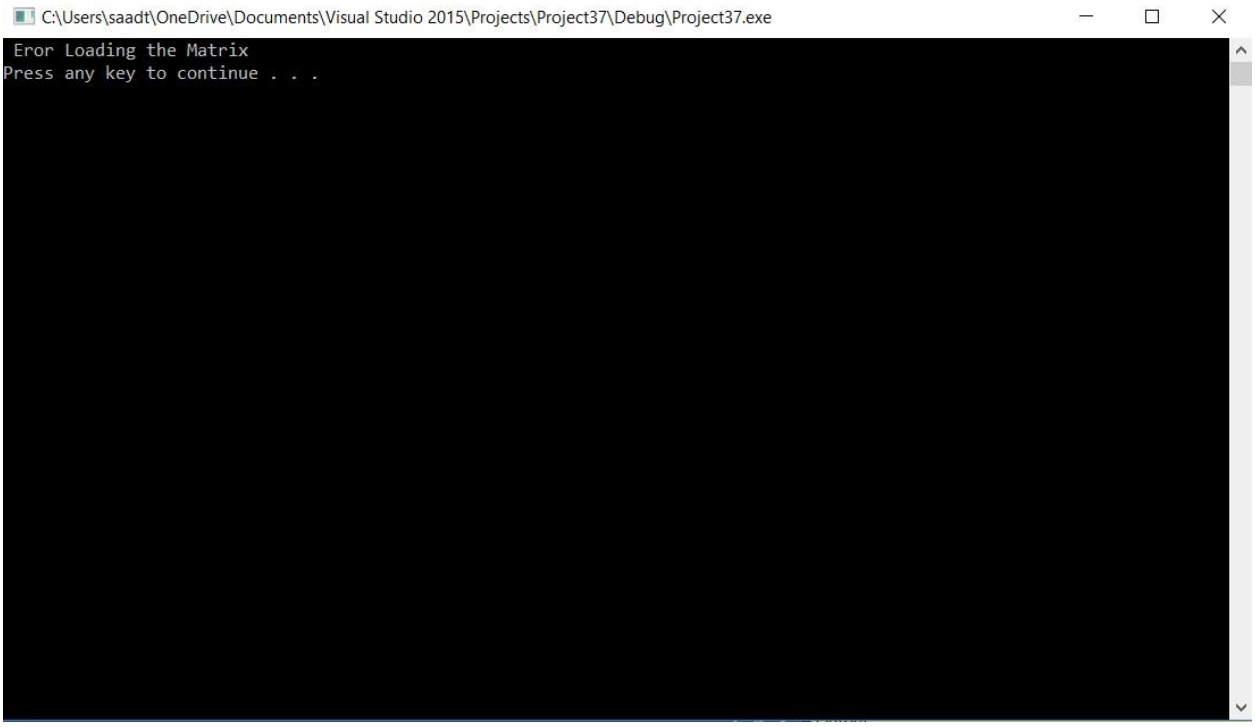


Test Case 12

Test Case 13

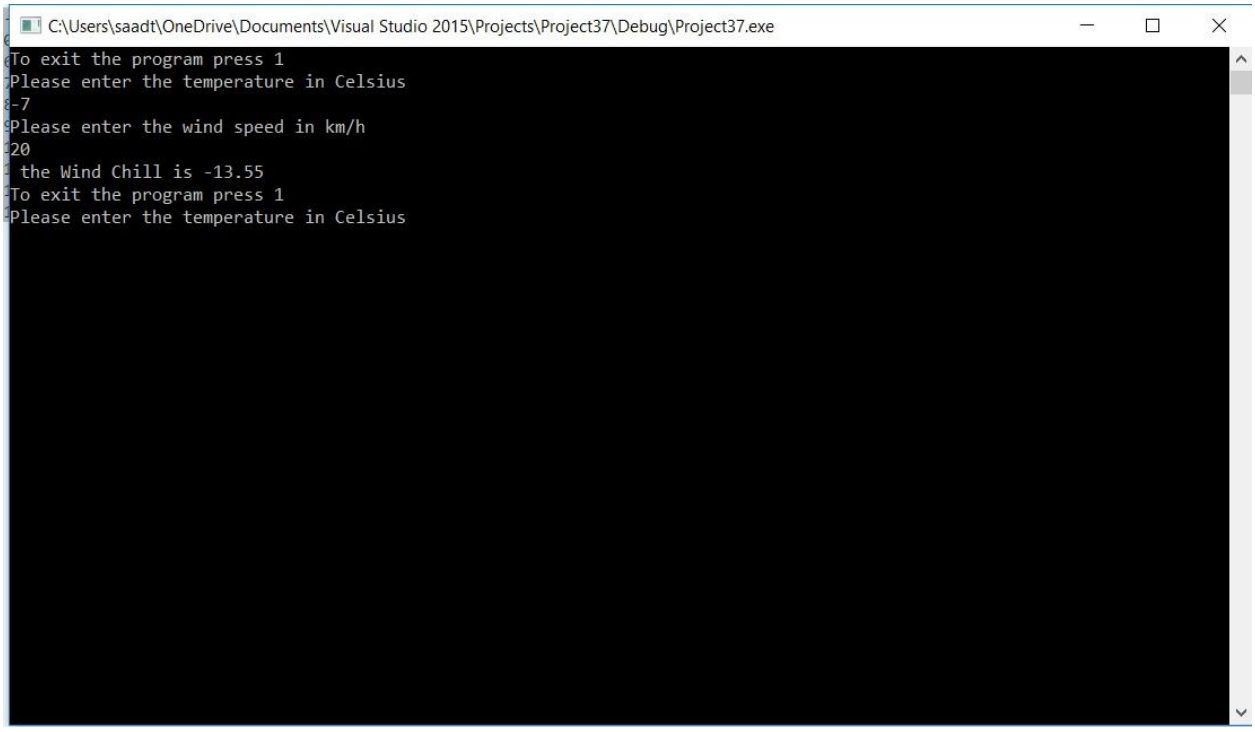


Test Case 14

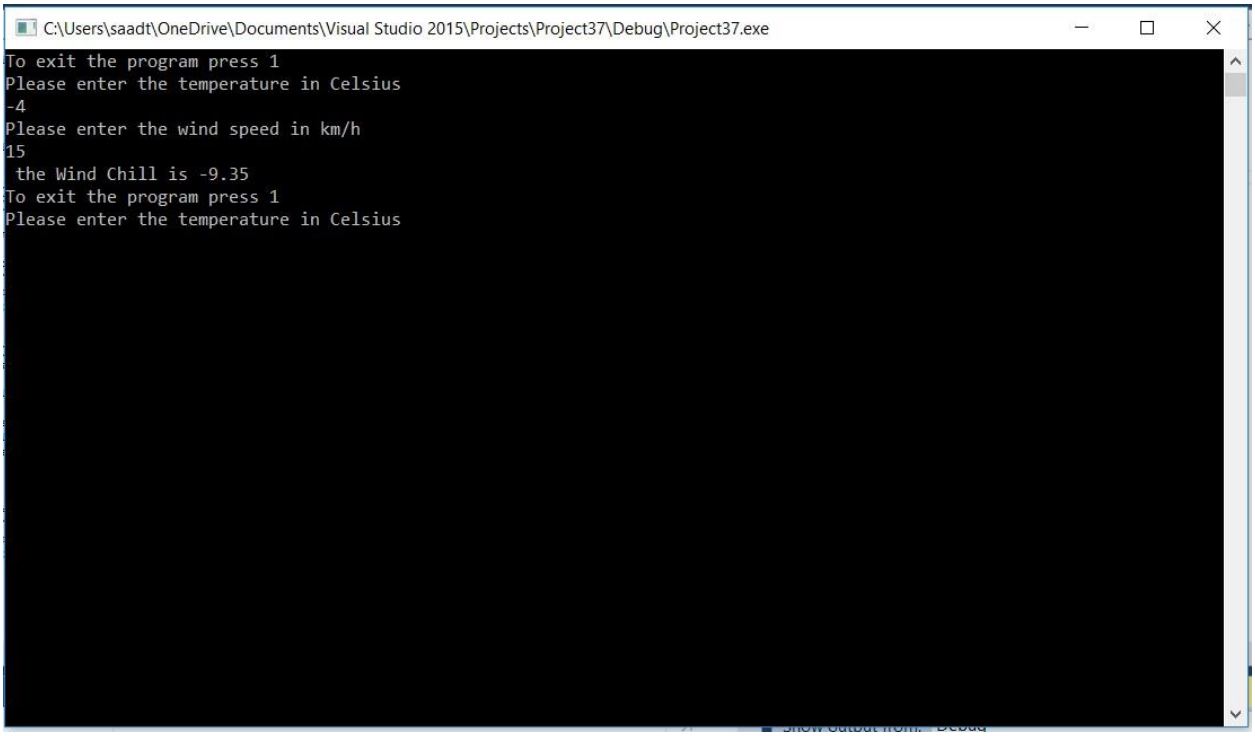


Test Case 15, 16, 17, 18.
The file will crash and will not open.

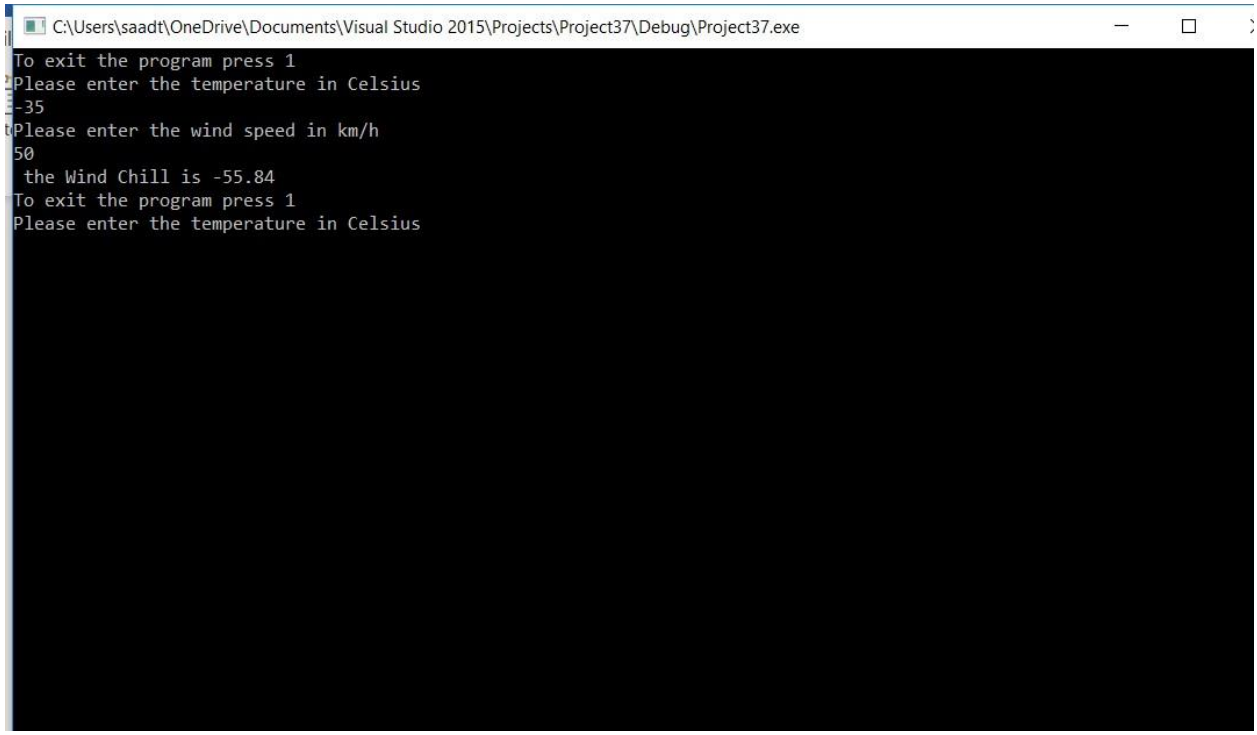
Test Case 19:



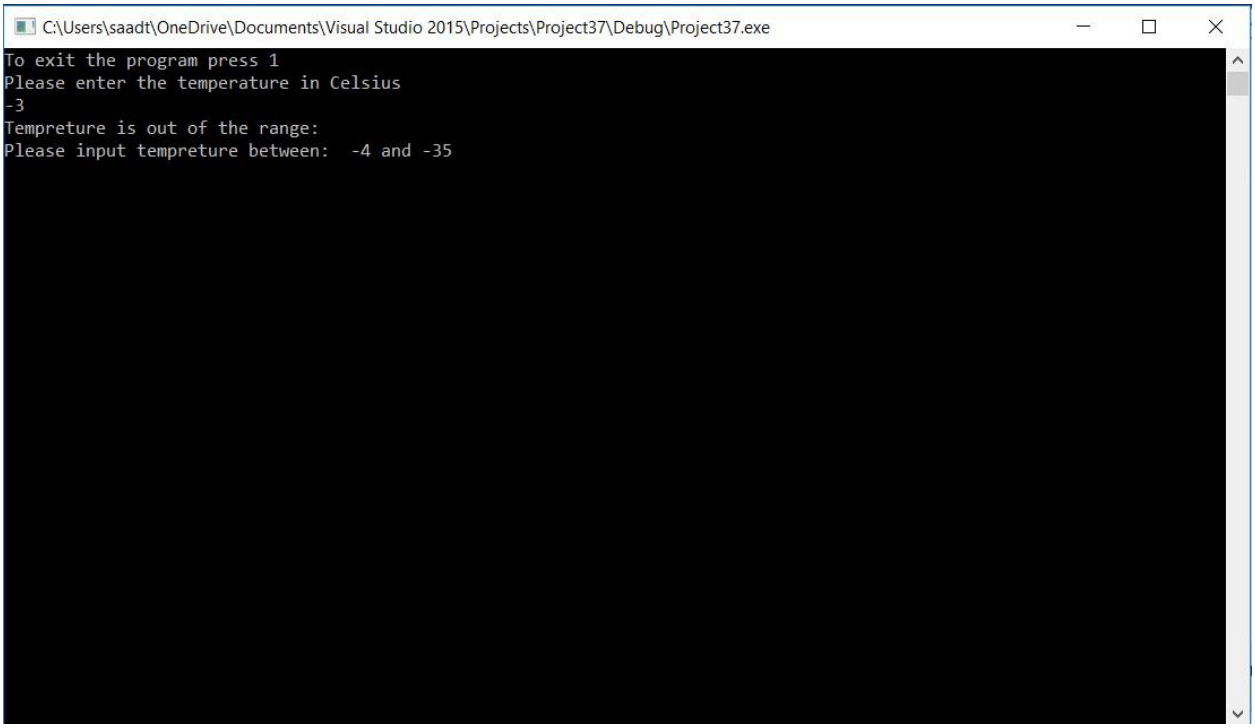
Test case 20:



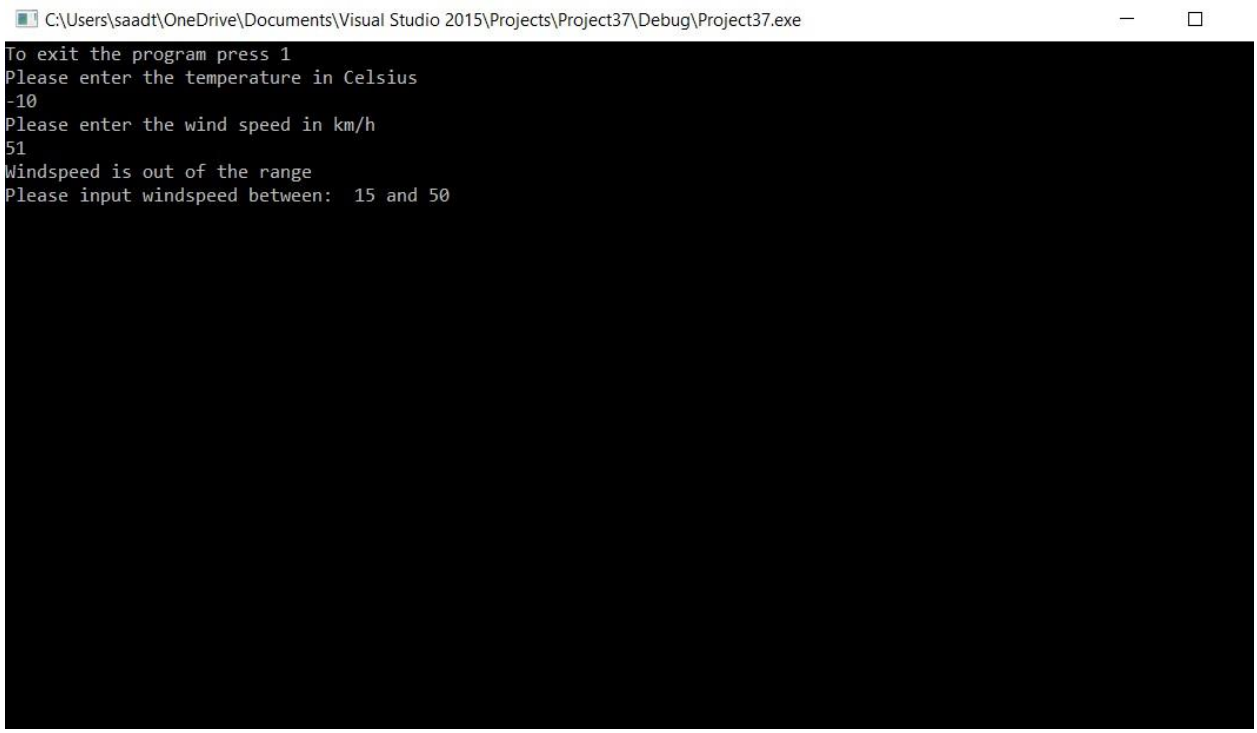
Test Case 21:



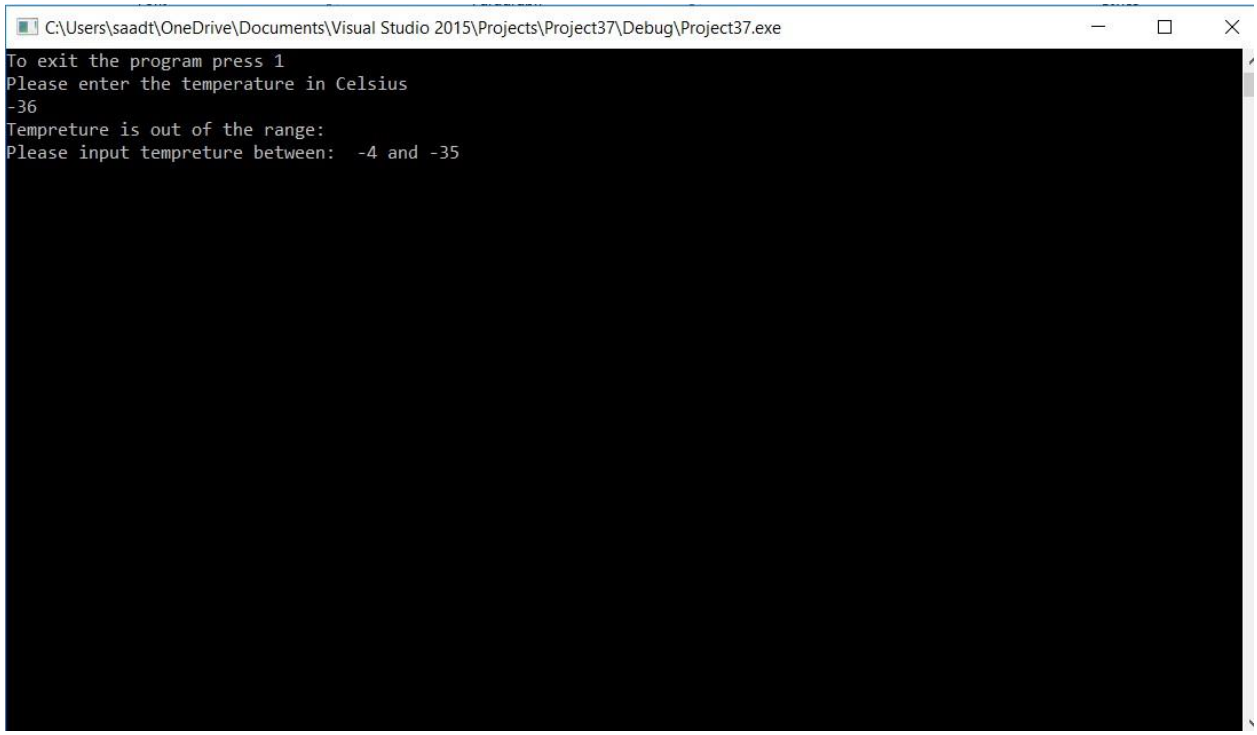
Test Case 22



Test case 23


C:\Users\saadt\OneDrive\Documents\Visual Studio 2015\Projects\Project37\Debug\Project37.exe
To exit the program press 1
Please enter the temperature in Celsius
-10
Please enter the wind speed in km/h
51
Windspeed is out of the range
Please input windspeed between: 15 and 50

Test Case 24


C:\Users\saadt\OneDrive\Documents\Visual Studio 2015\Projects\Project37\Debug\Project37.exe
To exit the program press 1
Please enter the temperature in Celsius
-36
Tempreture is out of the range:
Please input tempreture between: -4 and -35