**Name: Muhammad Rizwan Khalid**

**Class: BSCS-6A**

**Reg. : 180459**

**Task: 01:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

void ifPrime();

int main()

{

ifPrime();

system("pause");

}

void ifPrime()

{

int num;

int flag;

printf("Enter any number: ");

scanf("%d", &num);

for (int i = 2; i <num; i++){

if (num % i == 0){

flag = 0;

}

}

if (flag == 0){

printf("This is not a prime number\n");

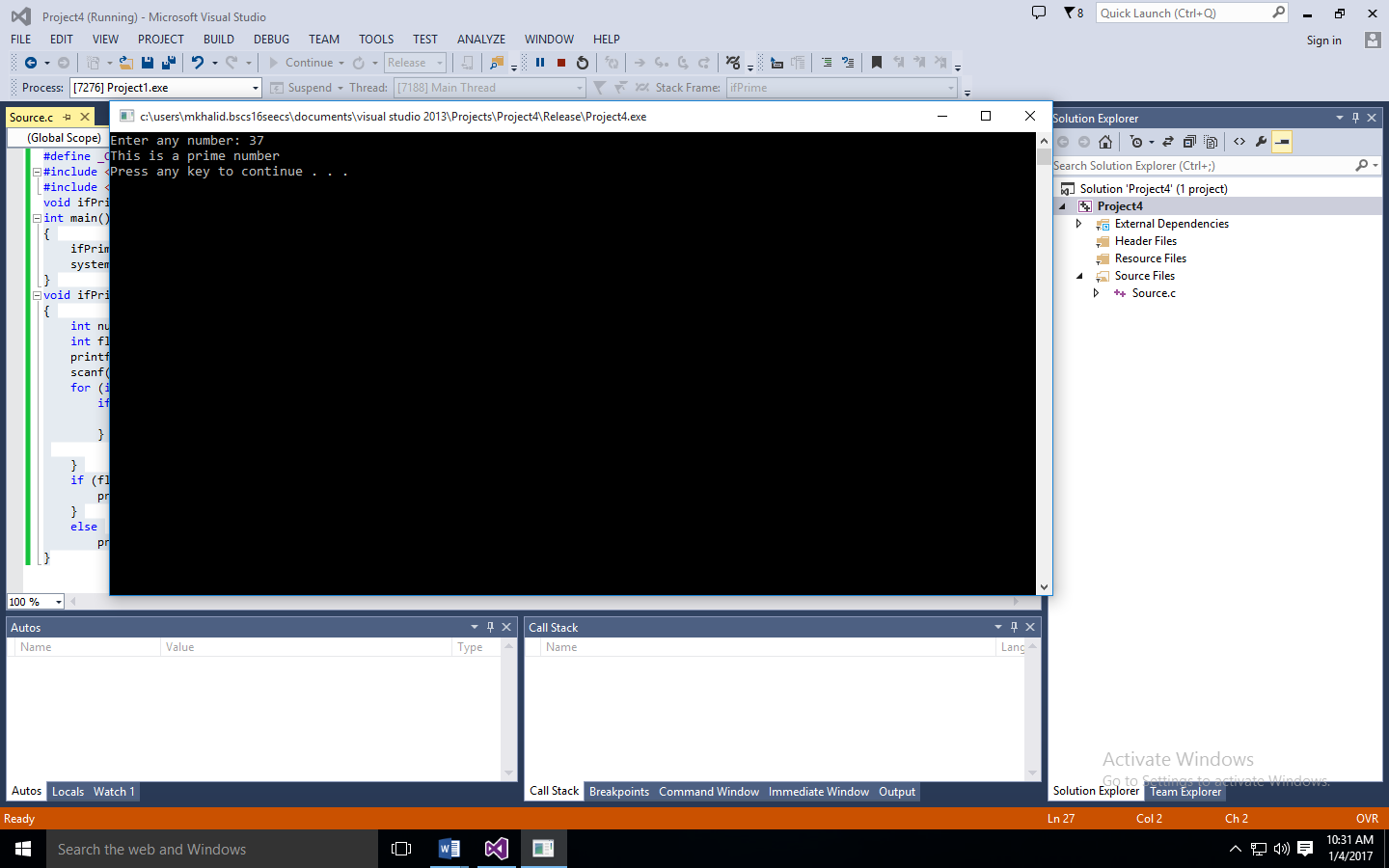
}

else

printf("This is a prime number\n");

}

**Output:**



**Task: 02: Minimum function: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

float Min(float, float);

int main()

{

float a, b, c, result;

printf("Enter first number: ");

scanf("%f", &a);

printf("Enter second number: ");

scanf("%f", &b);

printf("Enter third number:");

scanf("%f", &c);

result = Min(Min(a, b), c);

printf("\nThe minimum Number is %.3f\n", result);

system("pause");

}

float Min(float a, float b)

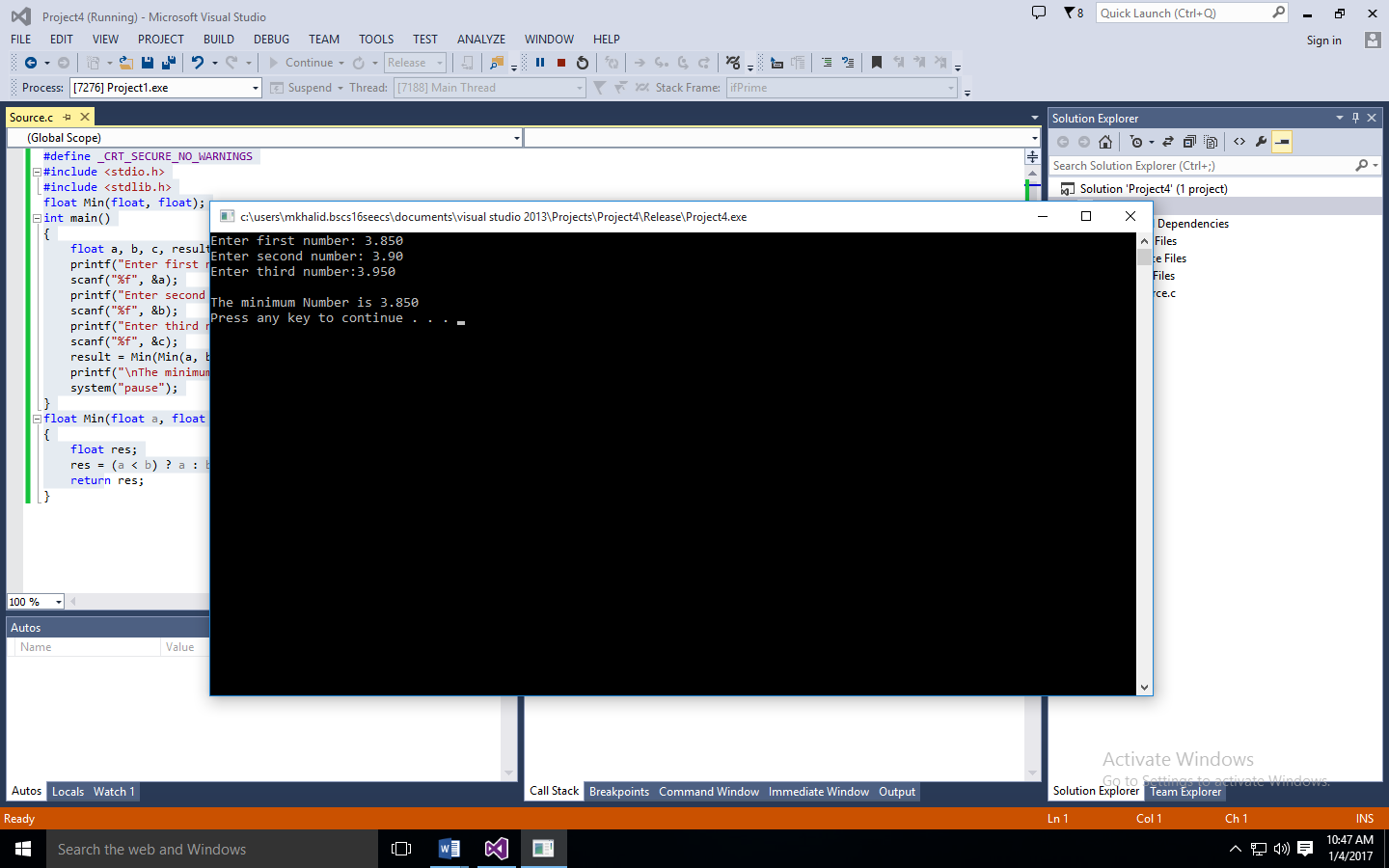
{

float res;

res = (a < b) ? a : b;

retur

**Output: -**



**Task: 02: Perfect number:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

int perfect(int);

int main()

{

int num, result;

printf("Enter any number: ");

scanf("%d", &num);

result = perfect(num);

if (result == 1){

printf("This is a perfect number because sum of its factor is equal to the number.\n");

}

else

printf("This is not a perfect number because sum of its factor is not equal to the number.\n");

system("pause");

}

int perfect(int num)

{

int arr[10000];

int i, o, sum = 0;

for (i = 1; i <= num; i++){

if (num % i == 0){

arr[i] = i;

}

}

for (o = 1; o <= i; o++){

sum = sum + arr[o];

}

if (num == sum){

return 1;

}

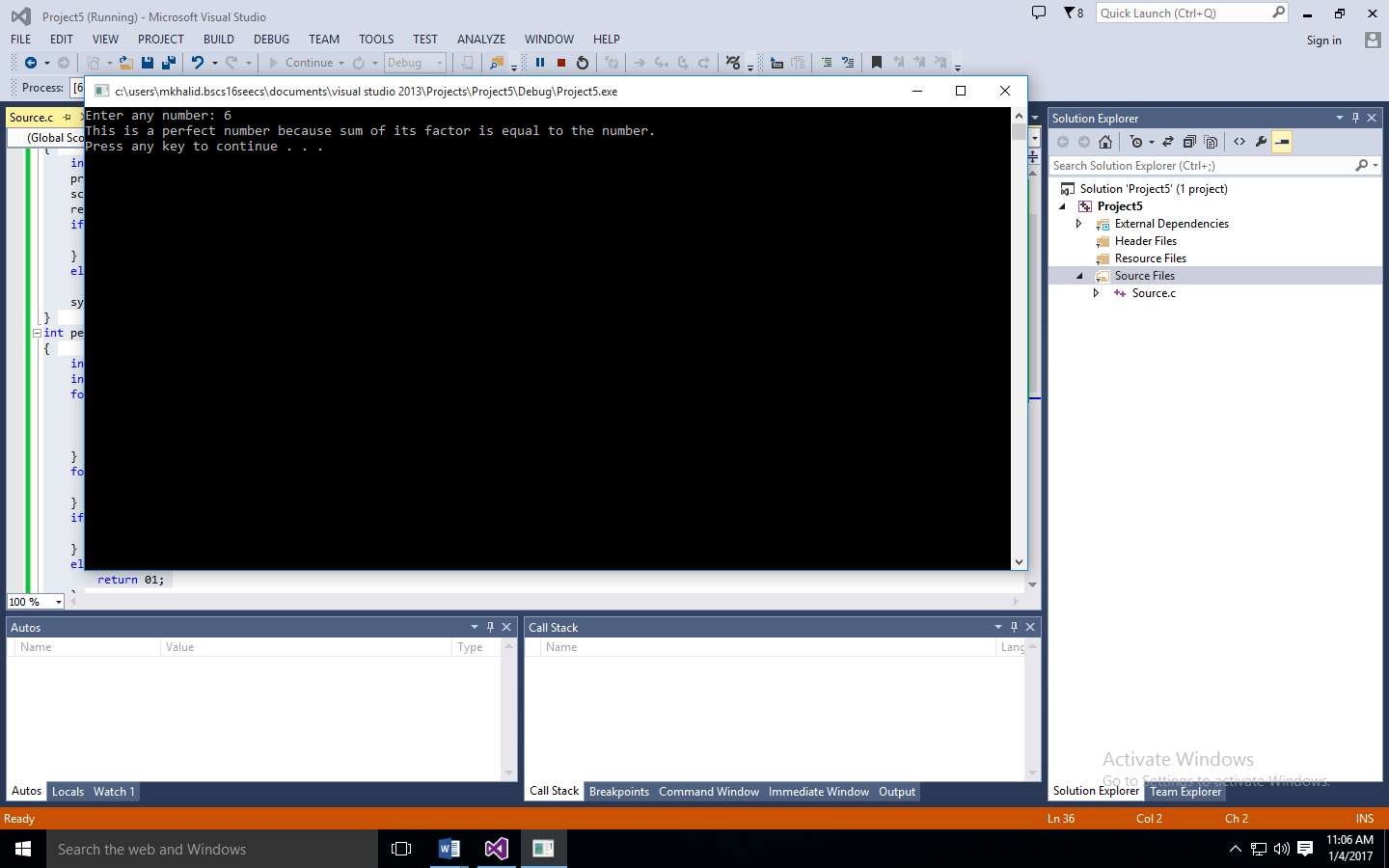
else {

return 01;

}

}

**Output**



**Task: 04: Sum of natural numbers:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

void Sum(int);

void main()

{

int num;

printf("Enter to calculate first n natural numbers: ");

scanf("%d", &num);

Sum(num);

system("pause");

}

void Sum(int num)

{

int sum = 0;

for (int i = 1; i <= num; i++){

printf("%d ", i);

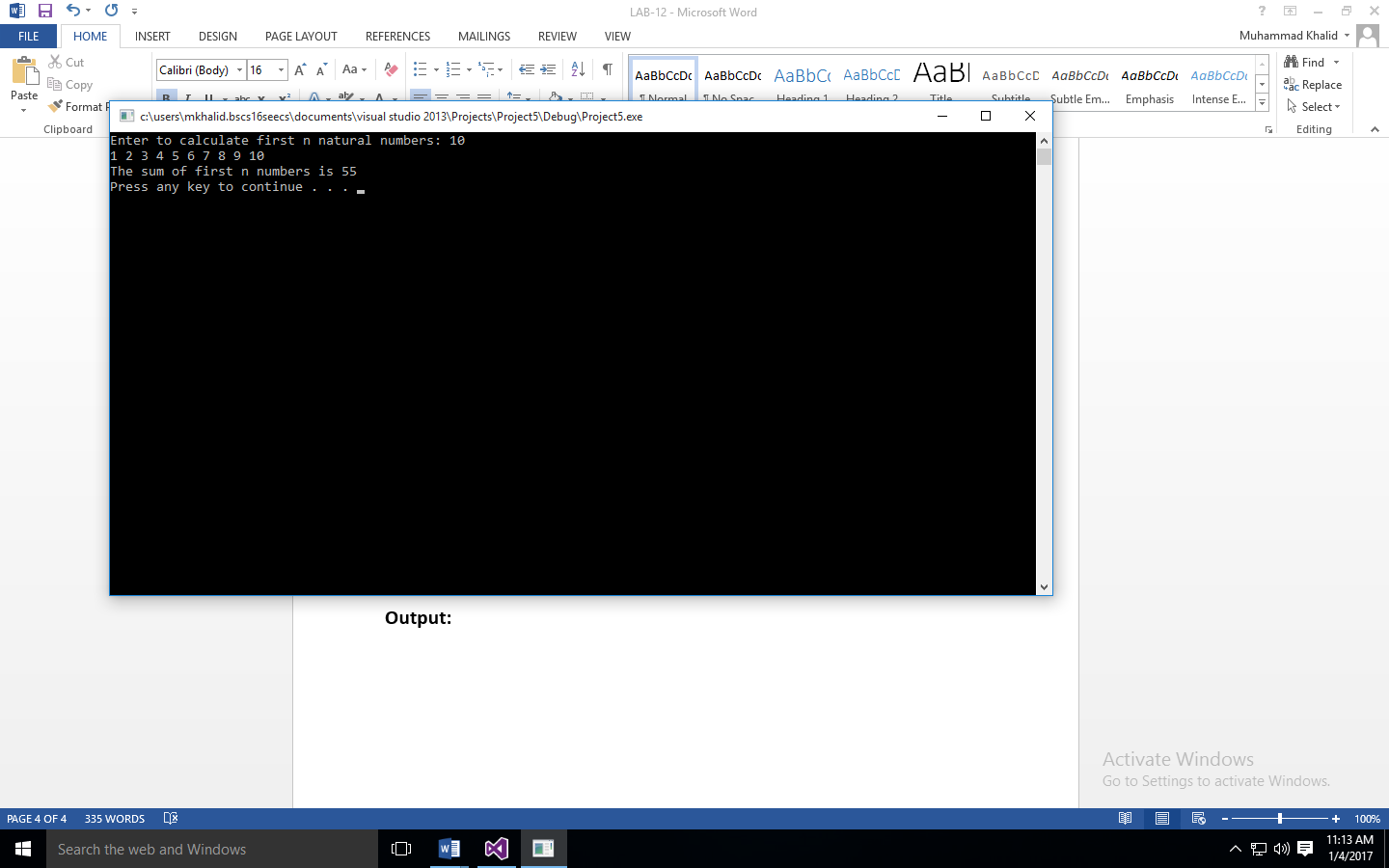
sum = sum + i;

}

printf("\nThe sum of first n numbers is %d \n", sum);

}

**Output:**



**Task: 05: Remove the errors: -**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

void Patern();

int main()

{

Patern();

Patern();

getchar();

getchar();

return 0;

}

void Patern()

{

int i, space, rows, k = 0, count = 0, count1 = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i)

{

for (space = 1; space <= rows - i; ++space)

{

printf(" ");

++count;

}

while (k != 2 \* i - 1)

{

if (count <= rows - 1)

{

printf("%d ", (i + k));

++count;

}

else

{

++count1;

printf("%d ", (i + k - 2 \* count1));

}

++k;

}

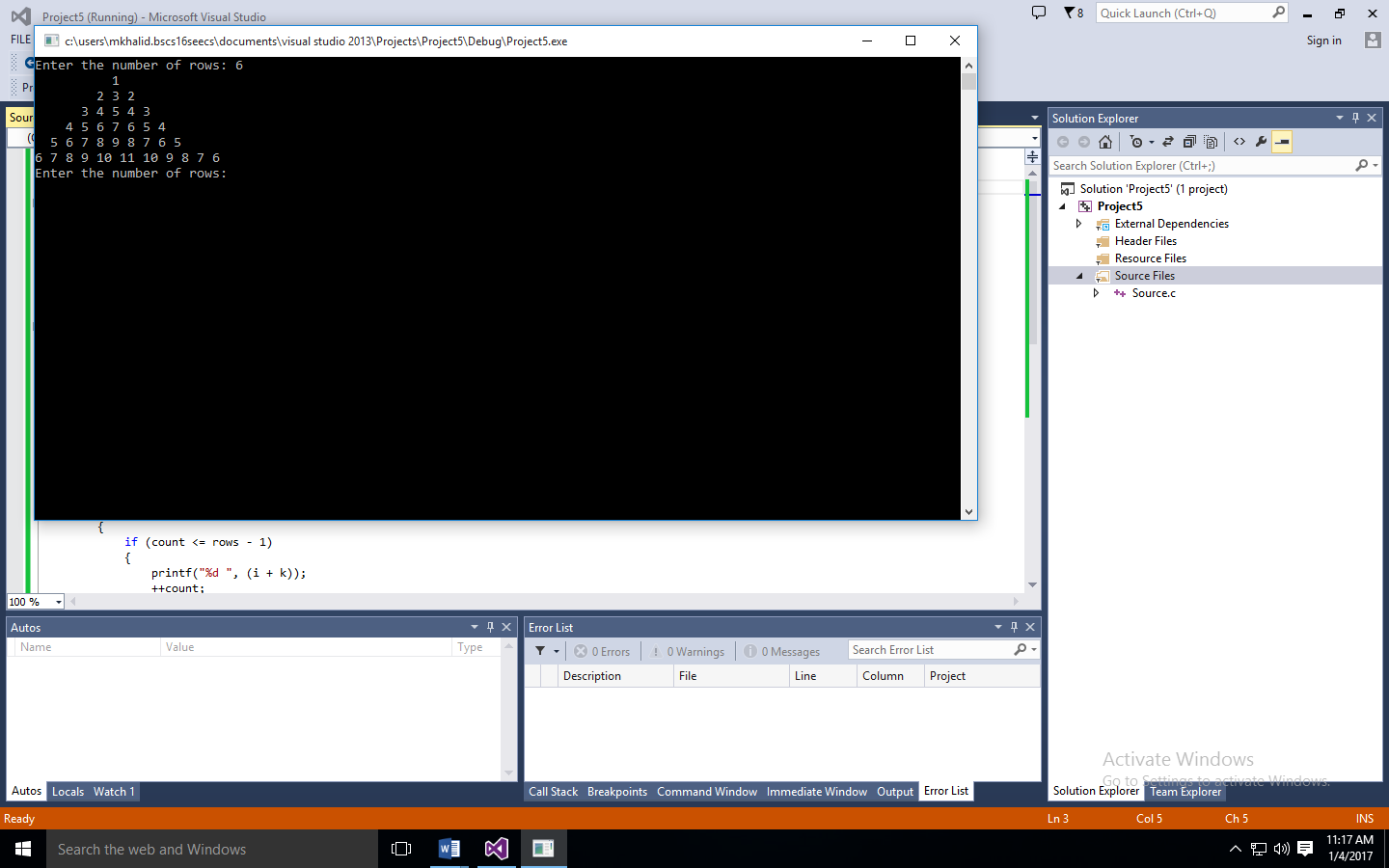
count1 = count = k = 0;

printf("\n");

}

}

**Output: -**



**Task: 06: Transforming the code into function: -**

/\*

C Program to Display the ATM Transaction

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

unsigned long amount = 1000, deposit, withdraw;

int choice, pin, k;

char transaction = 'y';

void Pin();

void operation();

void main()

{

Pin();

operation();

}

void Pin()

{

while (pin != 1520)

{

printf("ENTER YOUR SECRET PIN NUMBER:");

scanf("%d", &pin);

if (pin != 1520)

printf("PLEASE ENTER VALID PASSWORD\n");

}

}

void operation()

{

do

{

printf("\*\*\*\*\*\*\*\*Welcome to ATM Service\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("1. Check Balance\n");

printf("2. Withdraw Cash\n");

printf("3. Deposit Cash\n");

printf("4. Quit\n");

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

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("\n YOUR BALANCE IN Rs : %lu ", amount);

break;

case 2:

printf("\n ENTER THE AMOUNT TO WITHDRAW: ");

scanf("%lu", &withdraw);

if (withdraw % 100 != 0)

{

printf("\n PLEASE ENTER THE AMOUNT IN MULTIPLES OF 100");

}

else if (withdraw >(amount - 500))

{

printf("\n INSUFFICENT BALANCE");

}

else

{

amount = amount - withdraw;

printf("\n\n PLEASE COLLECT CASH");

printf("\n YOUR CURRENT BALANCE IS%lu", amount);

}

break;

case 3:

printf("\n ENTER THE AMOUNT TO DEPOSIT");

scanf("%lu", &deposit);

amount = amount + deposit;

printf("YOUR BALANCE IS %lu", amount);

break;

case 4:

printf("\n THANK U USING ATM");

break;

default:

printf("\n INVALID CHOICE");

}

printf("\n\n\n DO U WISH TO HAVE ANOTHER TRANSCATION?(y/n): \n");

fflush(stdin);

scanf("%c", &transaction);

if (transaction == 'n' || transaction == 'N')

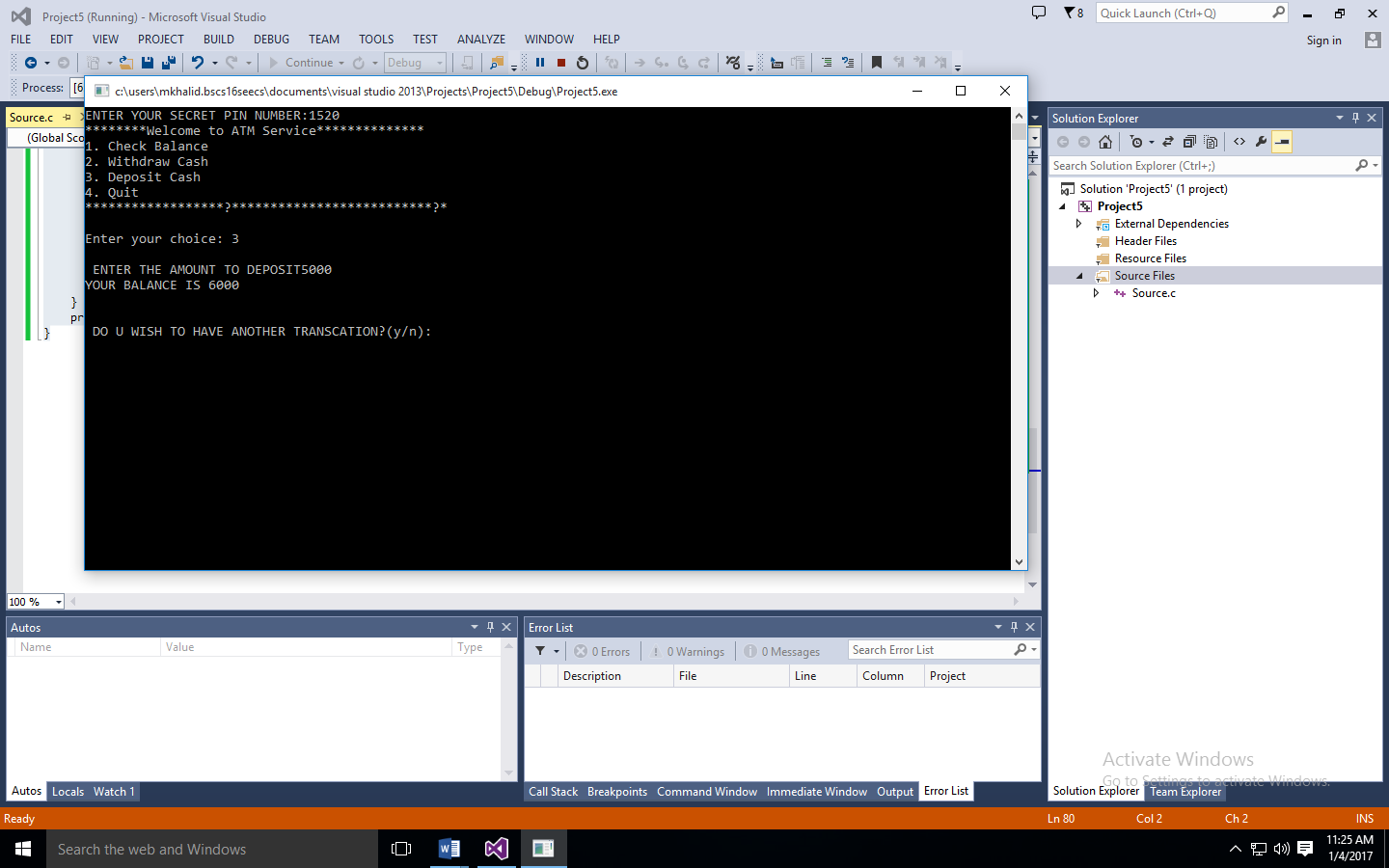
k = 1;

} while (!k);

printf("\n\n THANKS FOR USING OUT ATM SERVICE");

}

**Output:-**



**Task: 07: Biggest Entry Function:-**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

int biggestEntry(int x[2][3]);

void main()

{

int x[2][3] = { { 1, 2, 3 }, { 4, 7, 3 } };

int result;

result = biggestEntry(x);

printf("The biggest entry of array x[2][3] = { { 1, 2, 3 }, { 4, 7, 3 } } is %d\n ", result);

system("pause");

}

int biggestEntry(int x[2][3])

{

int a, b, max;

max = x[0][0];

for (a = 0; a < 2; a++){

for (b = 0; b < 3; b++){

if (x[a][b] > max){

max = x[a][b];

}

}

}

return max;

}

**Output: -**

