**דוח מעבדה 9**

**שמות מגישים:**

סער ויקטור – 312392822

אילון בן סימון – 312162951

**תרגיל 1**

קוד התכנית:  
header :

#ifndef example1

#define example1

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<math.h>

//functions declaration

double sum\_square(int m, int n, float(\*f)(int a));

#endif

Main:

#include"example1.h"

float func1(int x)

{

return (1.0 / x);

}

float func2(int x)

{

return (x / 5.0);

}

int main()

{

printf("The sum of func2: %.5lf\n", sum\_square(2, 13,func2));

printf("The sum of func1: %.5lf\n ", sum\_square(1, 10000,func1));

getch();

return 0;

}

Implementation:

#include"example1.h"

double sum\_square(int m, int n, float(\*f)(int a))

{

double sum = 0;

while (n >= m)//calculate the sum squre from n to m

{

sum += pow(f(n), 2);

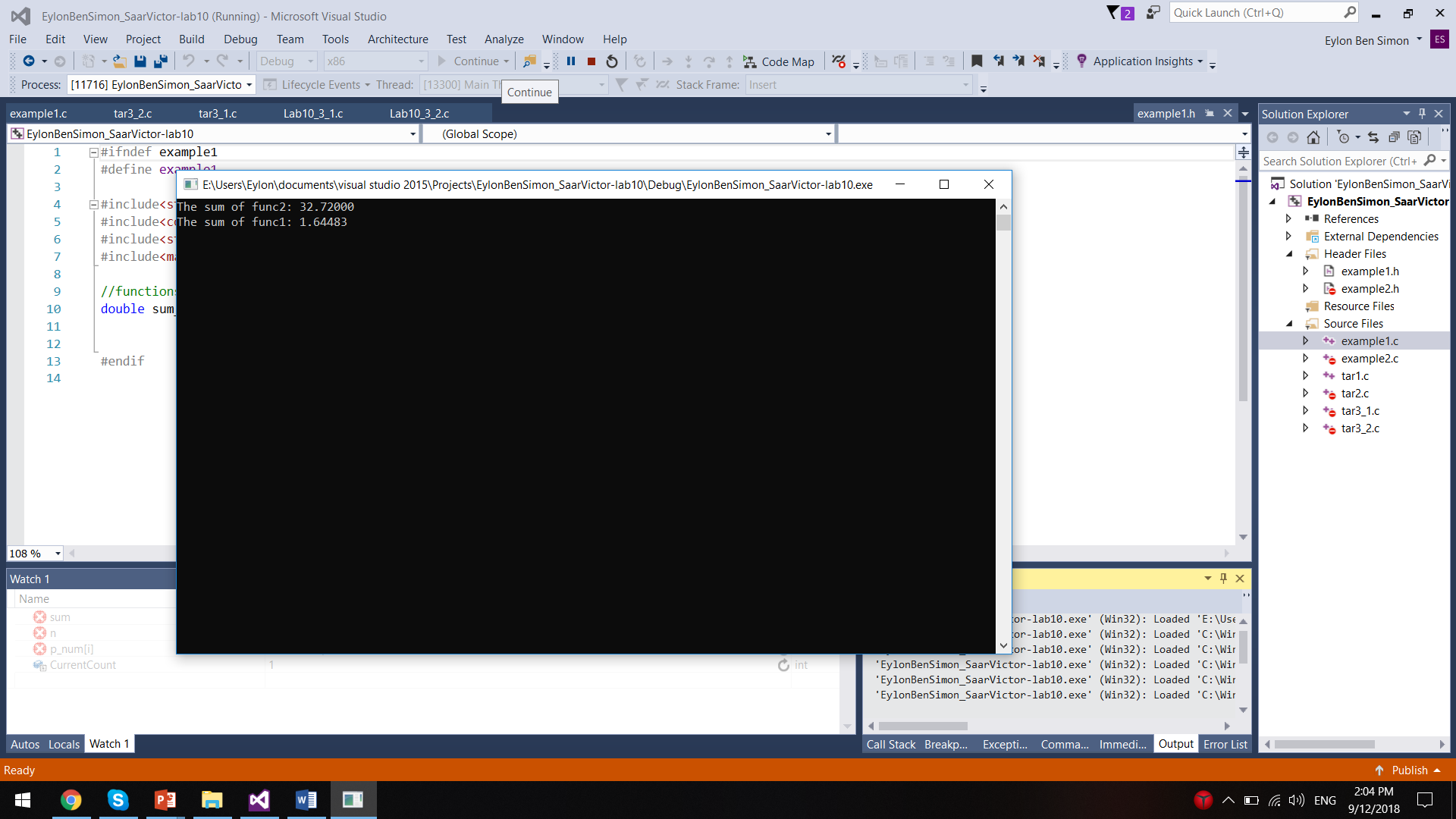
n--;

}

return sum;

}

פלט 1:



**תרגיל 2**

קוד התכנית:

header :

#ifndef example2

#define example2

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<math.h>

#define N 5

//functions declaration

typedef enum { FALSE, TRUE } BOOL;

BOOL Sum(BOOL(\*f)(void \*, void \*, void \*), void \*\*p\_num, void \*number);

#endif

Main:

#include"example2.h"

BOOL Int\_Sum(void \*a, void \*b, void \*c)

{

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

return TRUE;

return FALSE;

}

BOOL Float\_Sum(void \*a, void \*b, void \*c)

{

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

return TRUE;

return FALSE;

}

int main()

{

int num[] = { 3,5,23,5,6 }, i, value;

float fnum[] = { 3.5,5.0,2.3,5.8,6.2 }, fvalue;

void \*p\_num[N];

for (i = 0; i < N; i++)//fillings the pointers array

{

p\_num[i] = &num[i];

}

printf("\nPlease enter an integer number ");

scanf("%d", &value);

if (Sum(Int\_Sum, p\_num, &value) == TRUE)

printf("There is such sum\n");

else

printf("There is no such sum\n");

for (i = 0; i < N; i++)//fillings the pointers array

{

p\_num[i] = &fnum[i];

}

printf("\nPlease enter a float number ");

scanf("%f", &fvalue);

if (Sum(Float\_Sum, p\_num, &fvalue) == TRUE)

printf("There is such sum\n");

else

printf("There is no such sum\n");

getch();

return 0;

}

Implementation:

#include"example2.h"

//The function cheacks if the sum of two numbers is equal to third number

BOOL Sum(BOOL(\*f)(void \*, void \*, void \*), void \*\*p\_num, void \*number)

{

int i, j;

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

{

for (j = i+1; j <= N-1; j++)

{

if (f((p\_num)[i], p\_num[j], number) == TRUE)

return TRUE;

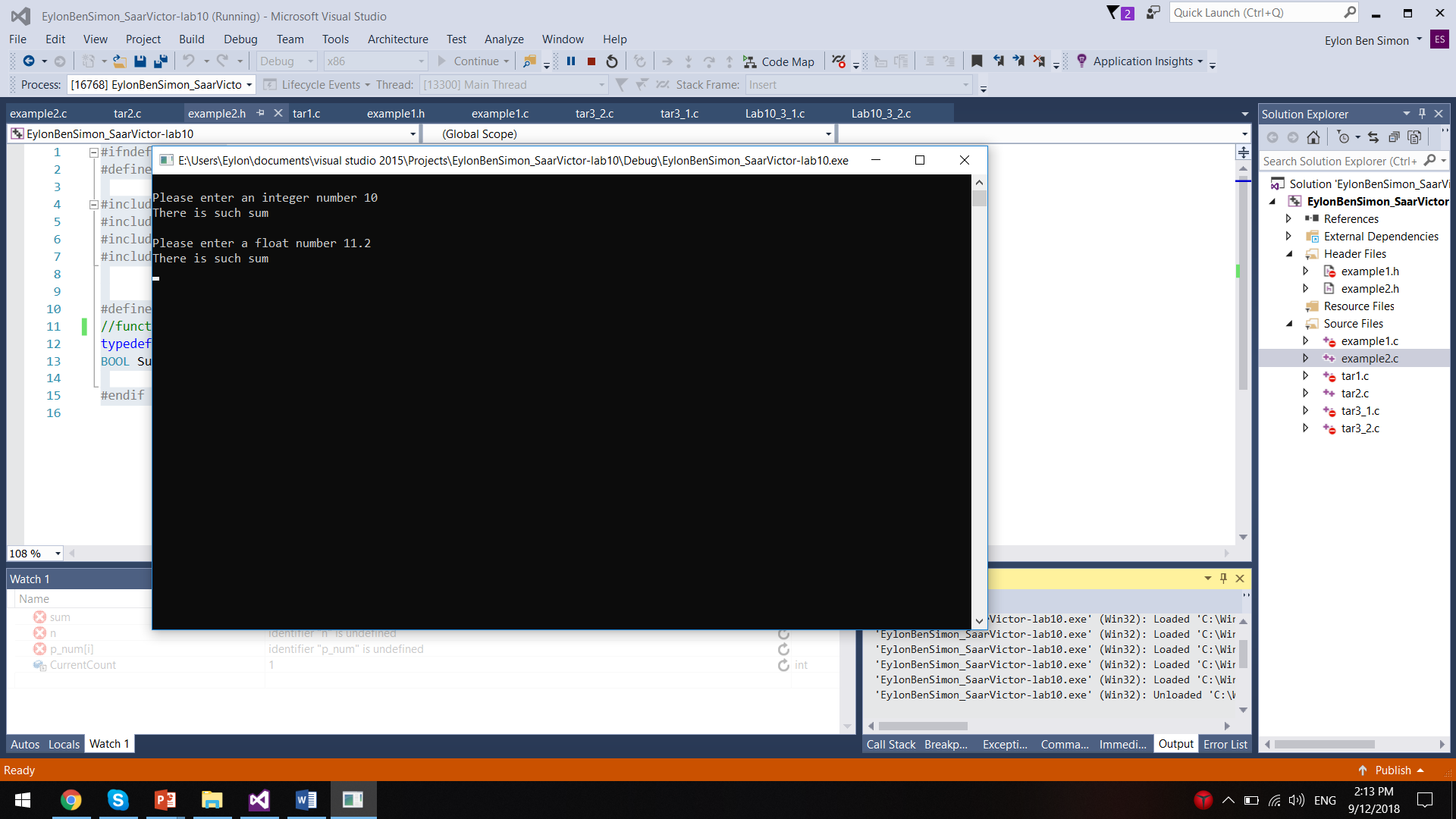
}

}

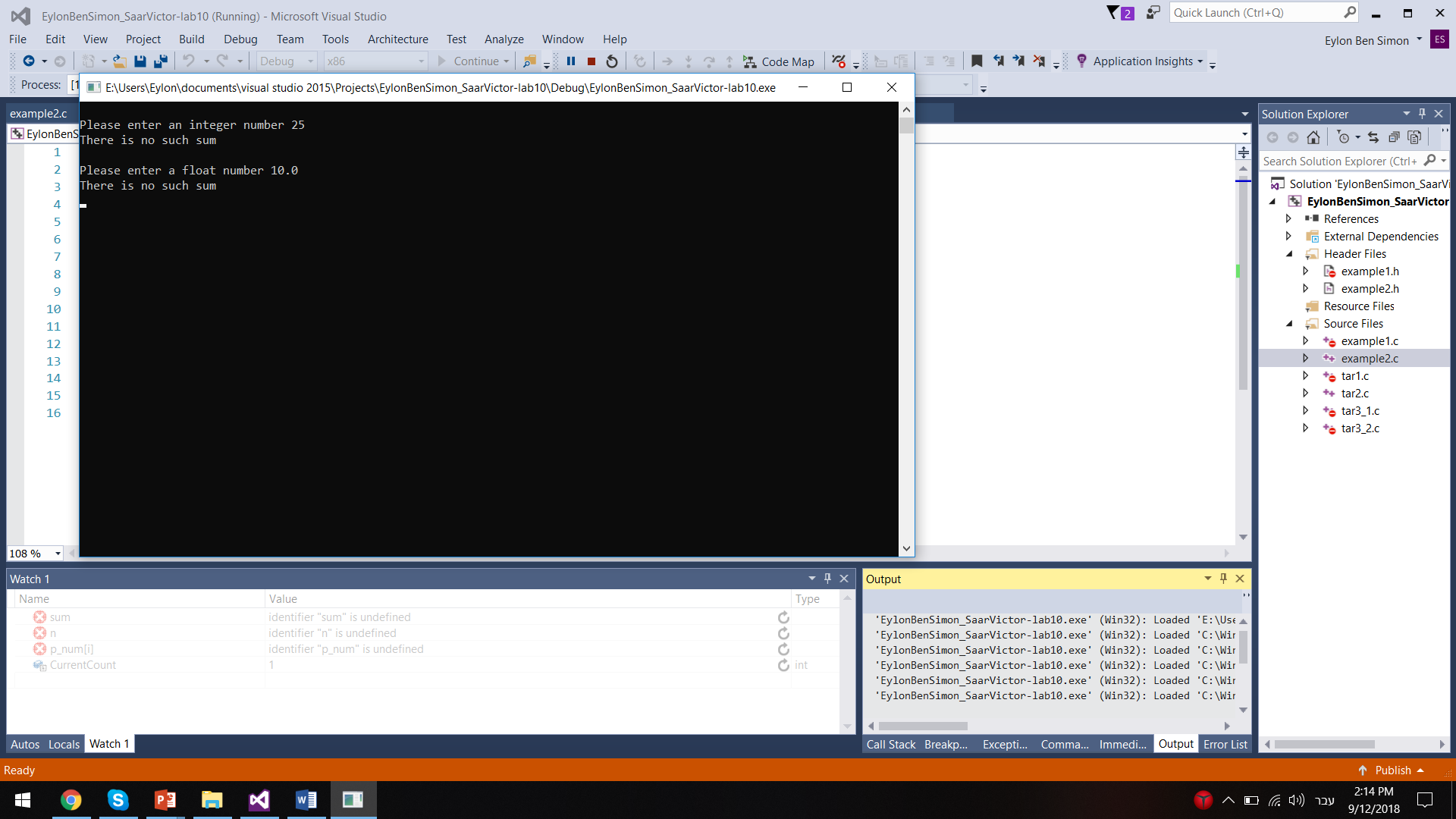
return FALSE;

}

פלט 1:



פלט 2:



**תרגיל 3**

קוד התכנית עבור int:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

#define N 5

typedef enum { FALSE, TRUE } BOOL;

void Insert(BOOL(\*f)(void \*, void \*), void \*Element, void \*\* Parray);

BOOL Int\_Comp(void \*FirstElement, void \*SecondElement);

BOOL IsFull();

int CurrentCount = 0; /\*global variable - only for the first ADT lab!Please,don't use!\*/

int main()

{

int Array[N], Value, i;

void\* PArray[N]; /\*pointers array\*/

printf("Enter numbers, -999 to stop");

scanf("%d", &Value);

while (Value != -999)

{

if (!IsFull())

{

Array[CurrentCount] = Value;

Insert(Int\_Comp, &Array[CurrentCount], PArray); /\*call the general function\*/

scanf("%d", &Value);

}

else

{

printf("The array is full");

break;

}

}

printf("\n Your sorted array is:\n"); /\*print the array in sorted order\*/

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

printf(" %d ", \*(int\*)PArray[i]);

getch();

return 0;

}

BOOL IsFull()

{

if (CurrentCount == N)

return TRUE;

return FALSE;

}

BOOL Int\_Comp(void \*FirstElement, void \*SecondElement)

{

if (\*(int\*)FirstElement > \*(int\*)SecondElement)

return TRUE;

return FALSE;

}

//The function insert the new element in the right place

void Insert(BOOL(\*f)(void \*, void \*), void \*Element, void \*\* Parray)

{

int i = CurrentCount - 1;

while(i>=0 && f(Element, Parray[i])==FALSE)//while elemnt<parray[i]

{

Parray[i + 1] = Parray[i];

i--;

}

Parray[i + 1] = Element;

CurrentCount++;

}

קוד התכנית עבור char:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

#define N 5

#define LENGTH 30

typedef enum { FALSE, TRUE } BOOL;

int CurrentCount = 0; /\*global variable - only for the first ADT lab!Please,don't use!\*/

void Insert(BOOL(\*f)(void \*, void \*), void \*Element, void \*\* Parray);

BOOL Str\_Comp(void \*FirstElement, void \*SecondElement);

BOOL IsFull();

int main()

{

char\* Array[N];

void\* PArray[N];

int i;

char TempStr[LENGTH];

printf("Enter strings with space between them,'end' to finish ");

scanf("%s", TempStr);

while (strcmp(TempStr, "end"))

{

if (!IsFull())

{

Array[CurrentCount] = (char\*)malloc(strlen(TempStr) + 1); /\*allocate the string memory\*/

strcpy(Array[CurrentCount], TempStr);

Insert(Str\_Comp, Array[CurrentCount], PArray); /\*call the general function\*/

scanf("%s", TempStr);

}

else

{

printf("The array is full");

break;

}

}

printf("\nYour sorted array is:\n");

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

printf(" %s ", (char\*)PArray[i]);

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

free(Array[i]);

getch();

return 0;

}

//The function insert the new element in the right place

void Insert(BOOL(\*f)(void \*, void \*), void \*Element, void \*\* Parray)

{

int i = CurrentCount - 1;

while (i >= 0 && f(Element, Parray[i]) == FALSE)//while elemnt<parray[i]

{

Parray[i + 1] = Parray[i];

i--;

}

Parray[i + 1] = Element;

CurrentCount++;

}

BOOL IsFull()

{

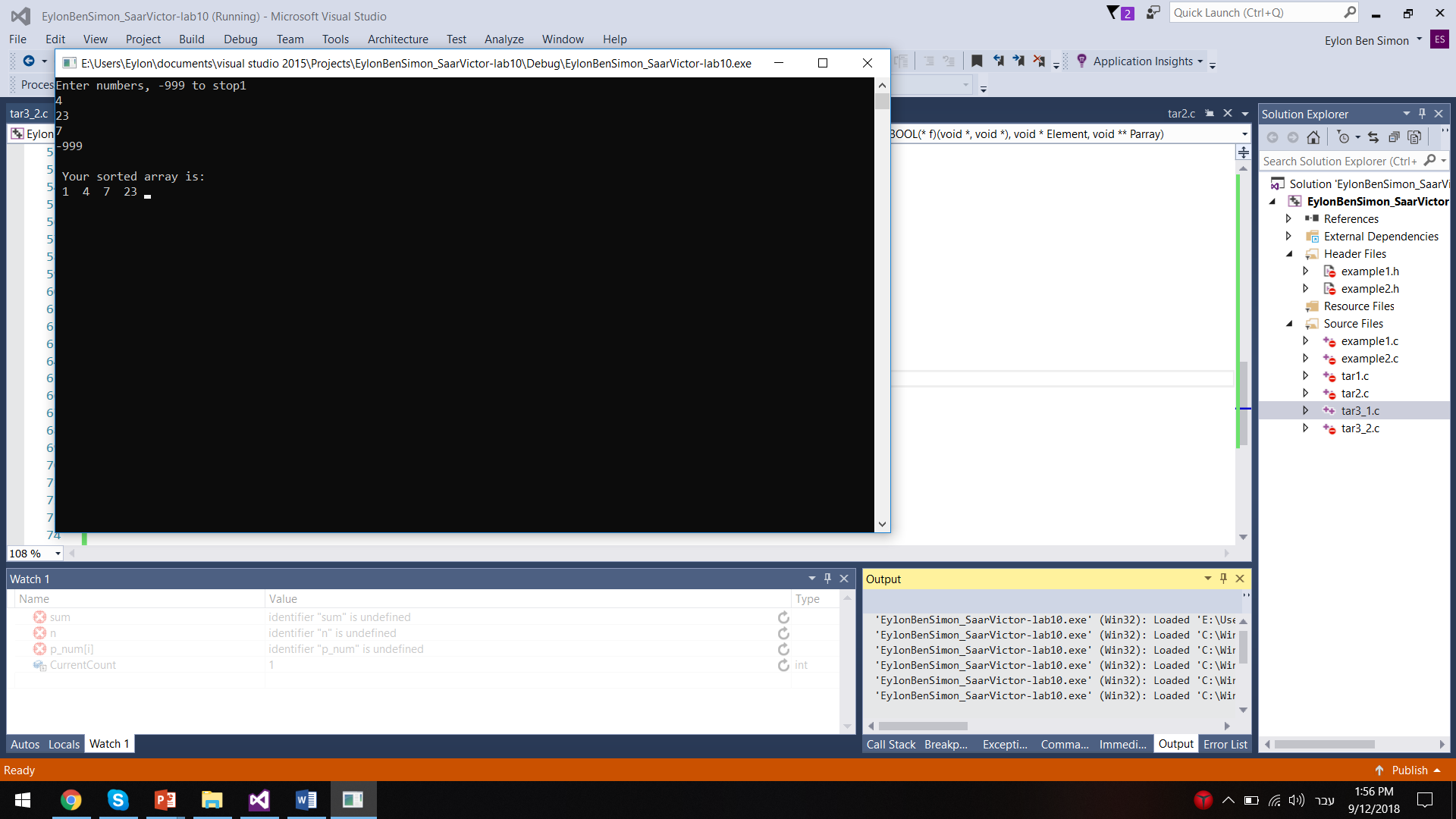
if (CurrentCount == N)

return TRUE;

return FALSE;

}

פלט 1:



פלט 2:

