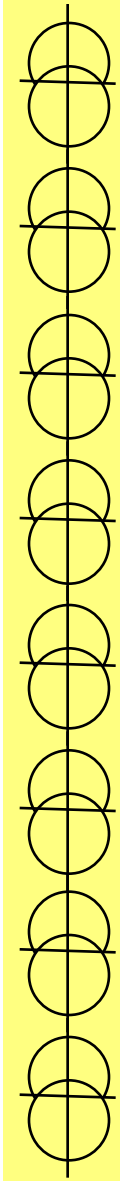
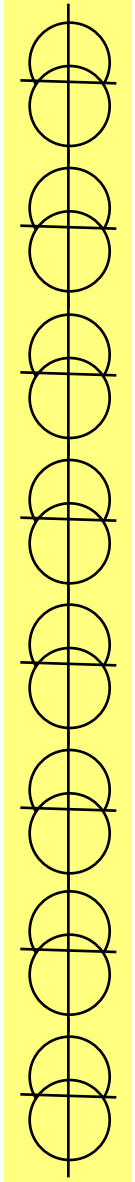


Tips and Tricks



Topics

- Visual Studio
- Handling Output to the Screen
- Handling Input from the Keyboard
- Simple Text Menus
- When to use a For Loop
- When to use a While Loop
- Reviewing a Function

Visual Studio

- Visual Studio is an extremely complex interactive development environment capable of handling many languages and tools.

How To Get Visual Studio?

1. Visual Studio is available to students at no charge through the DreamSpark program. Go to the link https://www.dreamspark.com/
2. Create your student account with university email id and login.
3. Go to software catalog and download Visual Studio Community 2015.

Visual Studio

- Deprecation-as languages develop and are updated over time, some commands are called “deprecated”, meaning they are not commonly supported.
- ANSI C is the root form of many versions of C, and some of its commands are “deprecated” in Visual Studio.

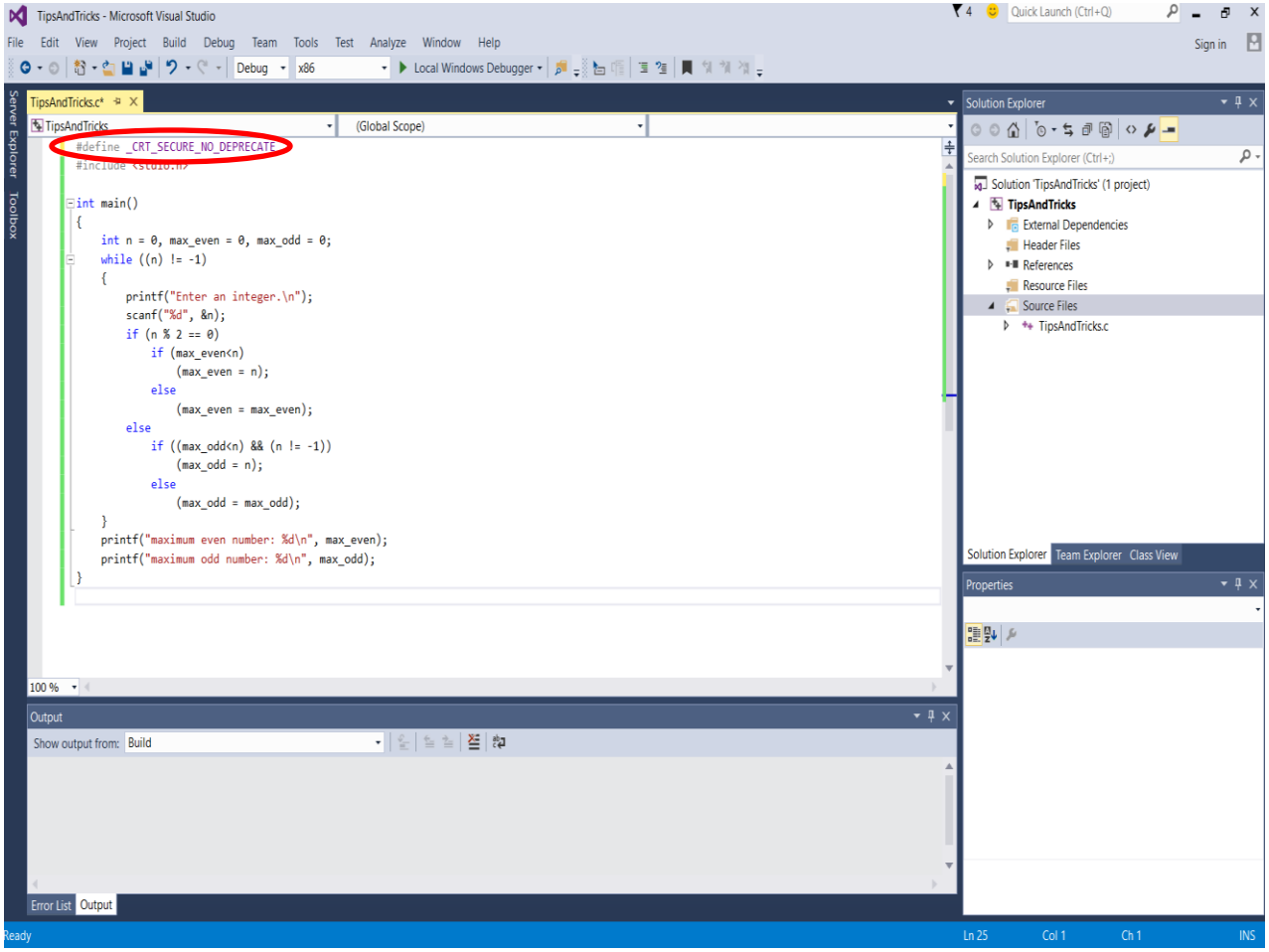
Visual Studio

- Visual Studio will create a “Warning or Error” for commands such as Scanf because of deprecation.
- The textbook, however, still uses these commands to instruct in the original ANSI C

Visual Studio

- To switch off the Deprecation Warnings in Visual Studio, a line must be added at the very beginning of the program.

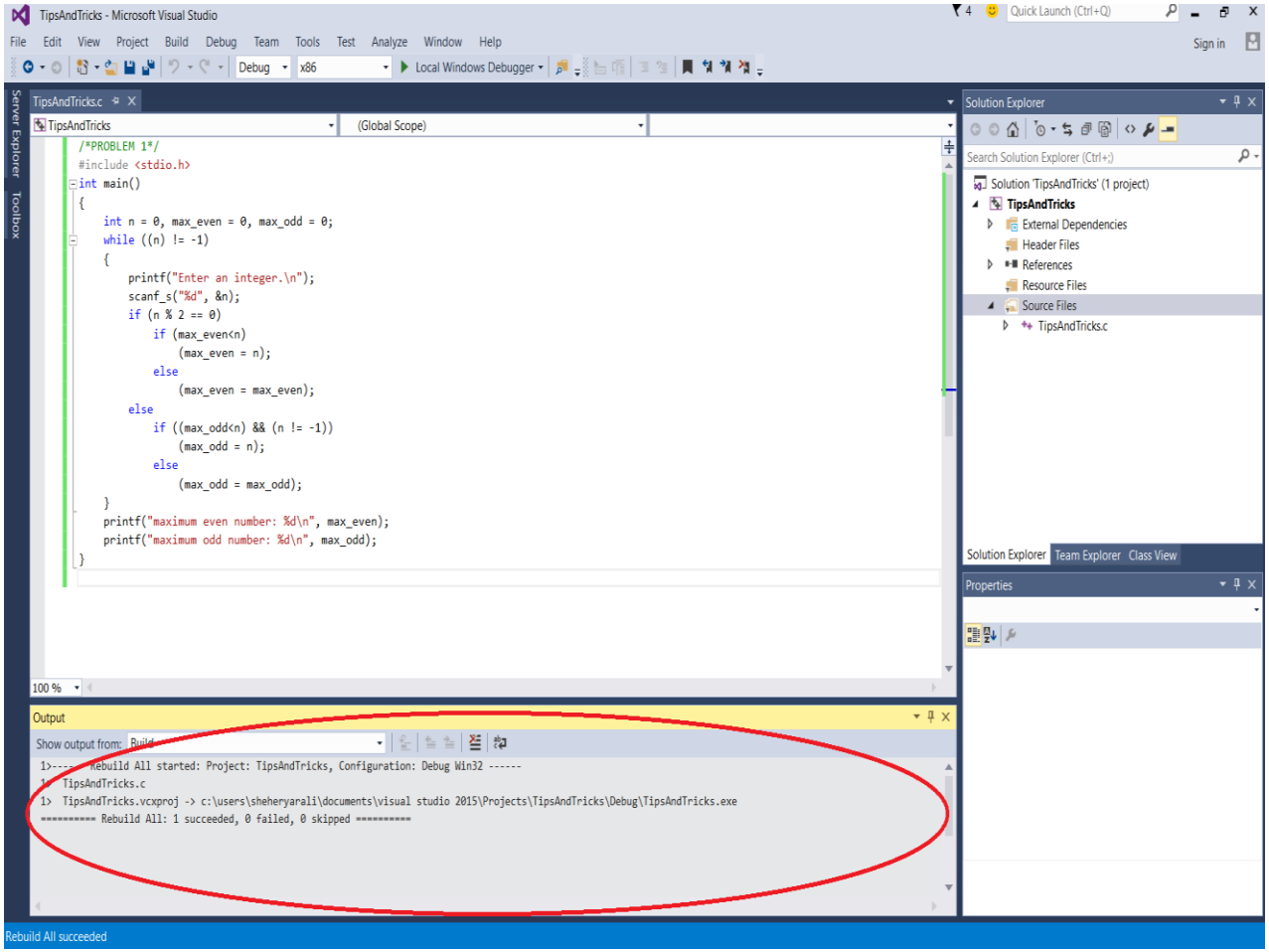
```
#define _CRT_SECURE_NO_DEPRECATE
```


TipsAndTricks - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
TipsAndTricks.c*
TipsAndTricks (Global Scope)
#define _CRT_SECURE_NO_DEPRECATE
#include <stdio.h>
int main()
{
int n = 0, max_even = 0, max_odd = 0;
while ((n) != -1)
{
printf("Enter an integer.\n");
scanf("%d", &n);
if (n % 2 == 0)
if (max_even<n)
(max_even = n);
else
(max_even = max_even);
else
if ((max_odd<n) && (n != -1))
(max_odd = n);
else
(max_odd = max_odd);
}
printf("maximum even number: %d\n", max_even);
printf("maximum odd number: %d\n", max_odd);
}
100 %
Output
Show output from: Build
Error List Output
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'TipsAndTricks' (1 project)
TipsAndTricks
External Dependencies
Header Files
References
Resource Files
Source Files
TipsAndTricks.c
Solution Explorer Team Explorer Class View
Properties
Server Explorer Toolbox
Ready
Ln 25 Col 1 Ch 1 INS

Visual Studio

- Another feature in Visual Studio is the reporting of errors in the output box.

Visual Studio



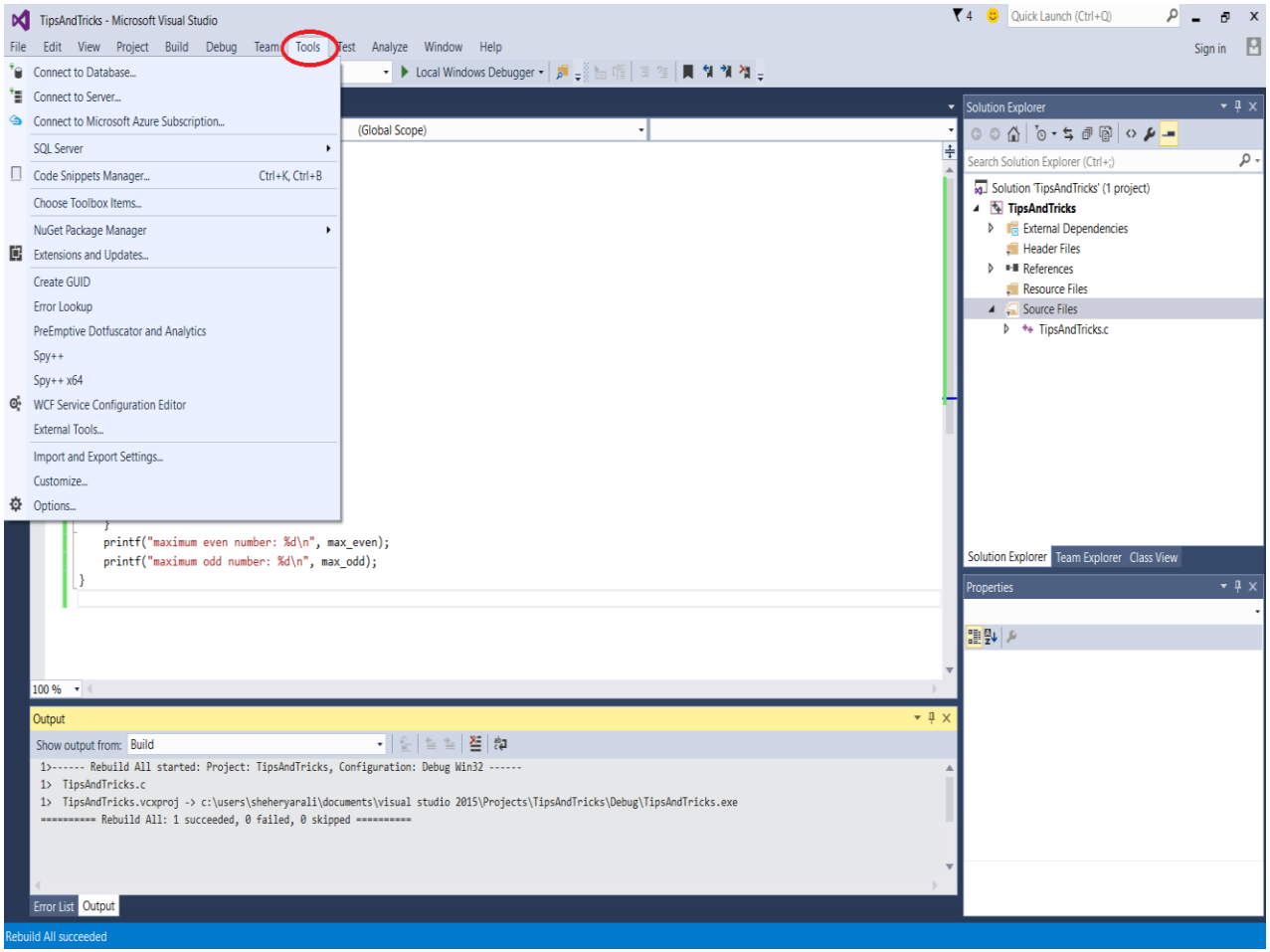
Visual Studio

- Clicking the error message in the output box usually causes the cursor to jump to the location of the error.
- An easier approach may be to look at the line numbers in the window.
- However, the line numbers may need to be turned on.

Turn on Line Numbers

- First, go to Tools on the Main Menu Bar. The Main Menu is always the menu bar at the top of a program

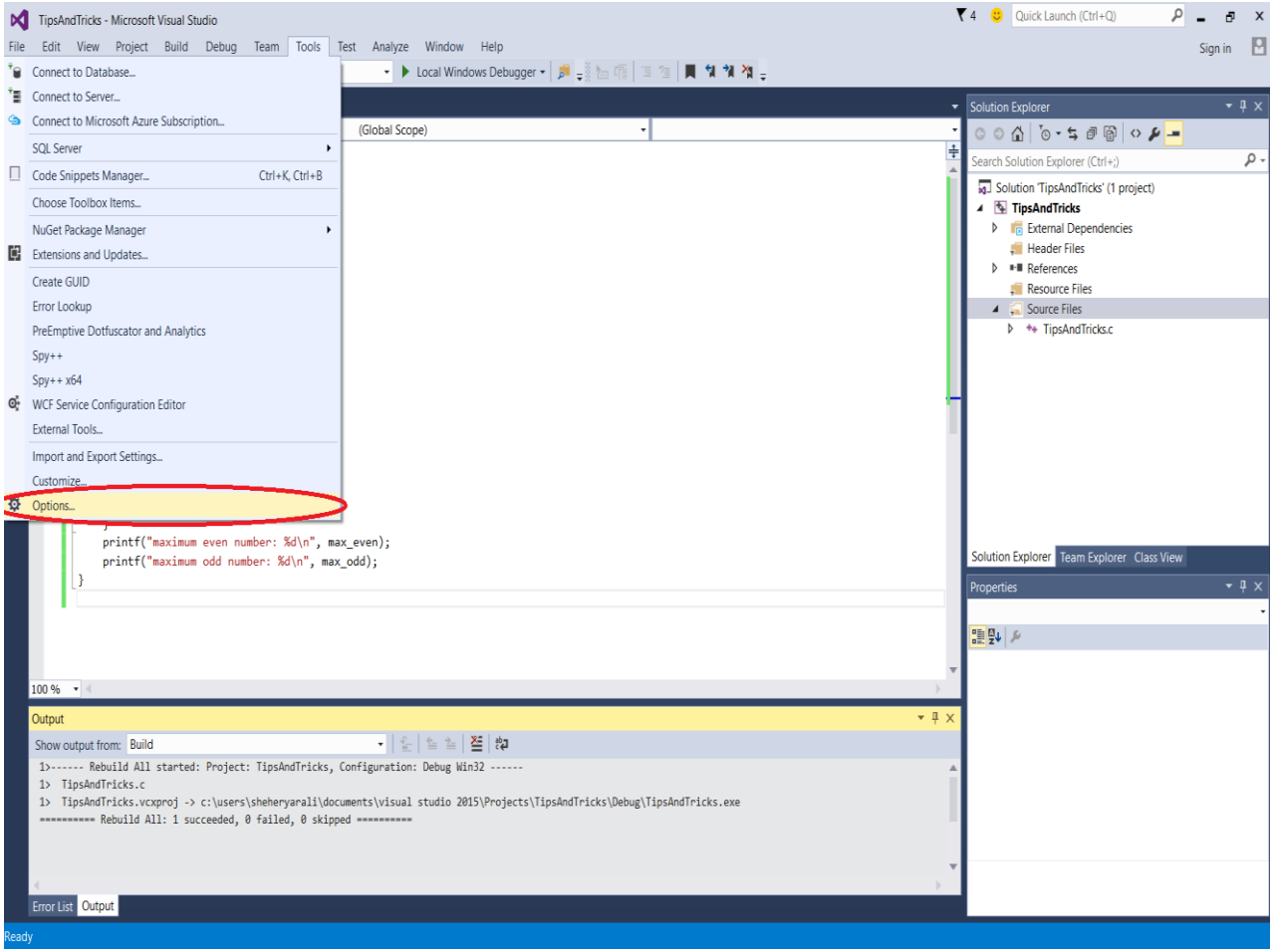
Visual Studio



Visual Studio

- Next, click Options at the bottom of the menu.

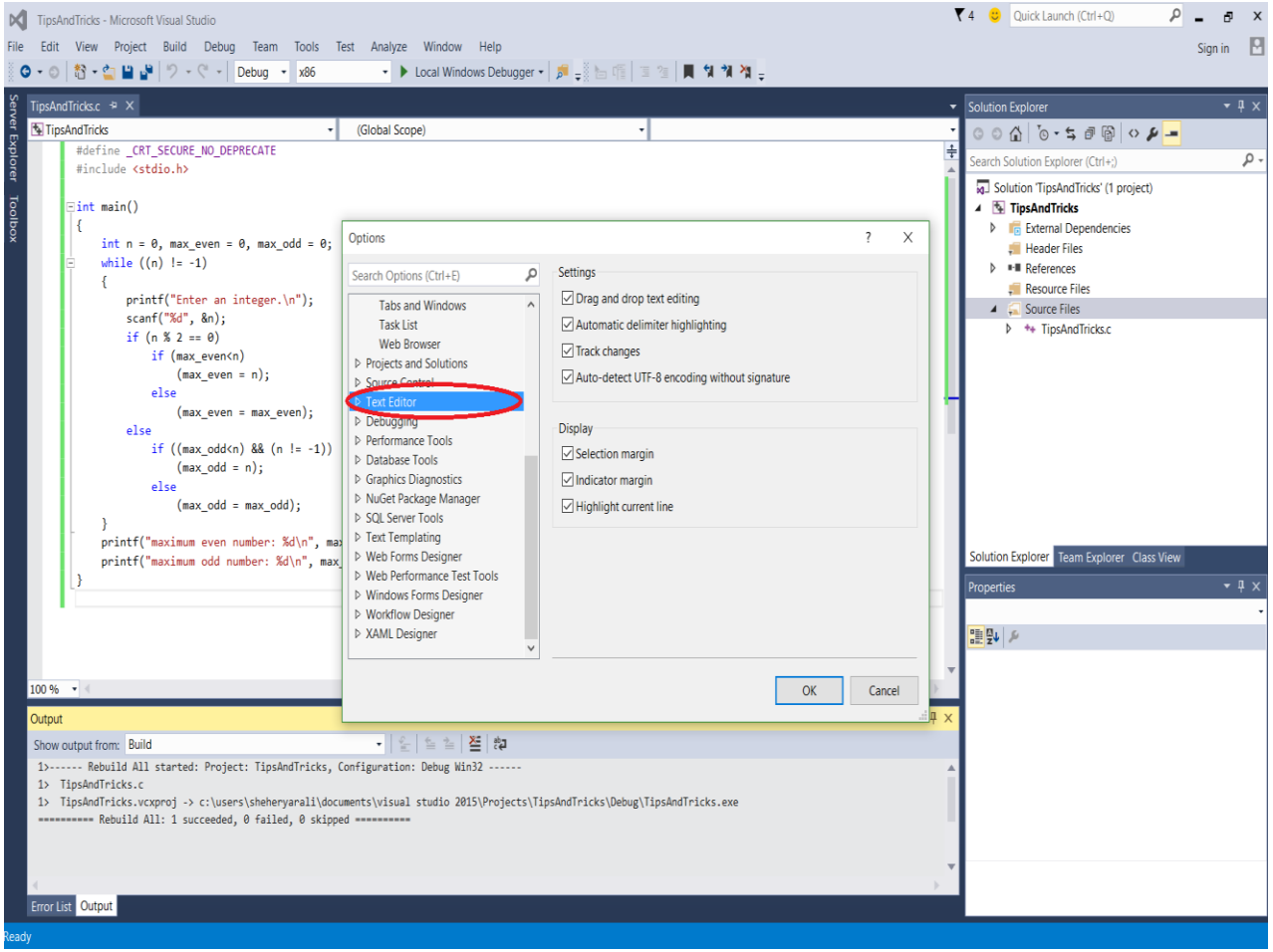
Visual Studio



Visual Studio

- Next the **Options Dialog Box** will appear. On the menu tree on the left, look for the words Text Editor and expand the options by selecting the “+” sign next to the entry.

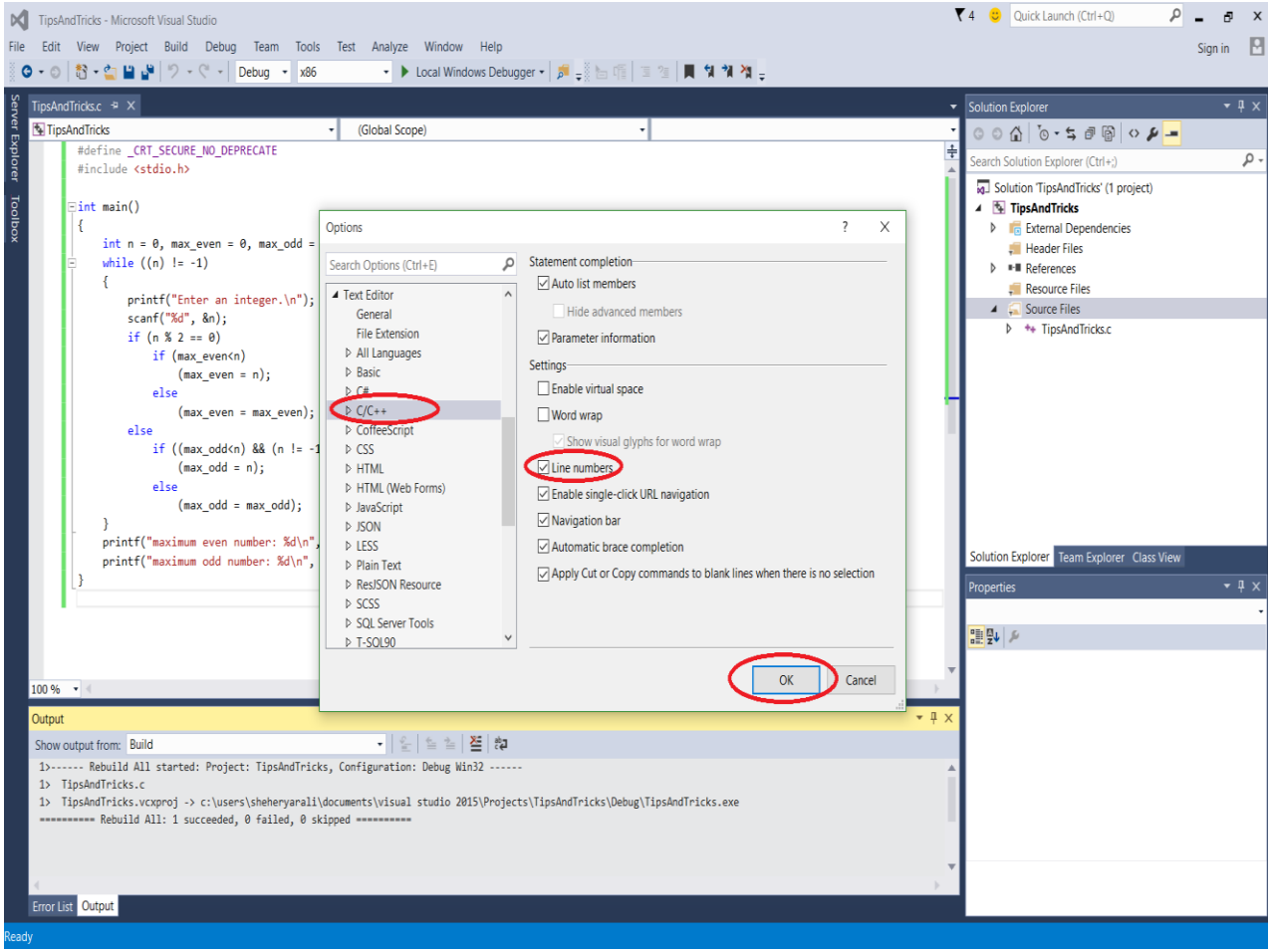
Visual Studio



Visual Studio

- Once expanded, many options will appear. Select the C/C++ entry on the tree.
- On the right side of the dialog box, several options will appear. Click the Line Number option.
- Click OK

Visual Studio



The Include Statement

- Reviewing from Chapter 1, the Include statement is useful.
- #include allows the program to have access to the header files of the system libraries.

Include

- #include <stdio.h> allows access to the main input and output routines,
  - scanf
  - printf
  - getchar

Include

- Another useful library is the console input output library.
- #include <conio.h>
  - _kbhit();
  - _getch();

Include

- #include <stdlib.h> allows access to the main library, which contains memory controls
  - Malloc-Allocate a single memory area
  - Calloc-Allocate a set of memory areas

Include

- #include <math.h> allows access to math routines, such as
  - sin-Sine
  - cos-Cosine
  - tan-Tangent
  - asin-Arcsine
  - acos-Arccosine
  - atan-Arctangent
  - pow-Power, x to the power of y
  - sqrt-Square Root

The Printf Statement

- All printf statements use a format string in order to display the output.
- A format string works by two conditions
- first, the type of data to be displayed
- second, the order of the data to be displayed.

The Printf Statement

- Each type of data has different combinations that can alter how it looks.
- Formats differ for Integers and Floats

The Printf Statement

- For Example, an integer can be shown as:
- %d-Simplest display
- %10d-At least 10 characters are shown, the remainder are spaces
- %010d, at least 10 characters are shown, the remainder are 0’s
- %+d, The sign of the integer will always be visible.
- And these can be combined.

The printf Statement

- For example, a float can be shown
- %f just prints the float as is.
- %6.3f-prints the float in 6 characters, with three characters reserved for the decimal portion.
- %+6.3f-uses one of the six characters for the sign, and 3 characters for the decimal
- %016.3f-As with integer, the leading spaces are replaced with zeros
- As before, these can be combined

Scanf

- Many times, when Scanf is used, the data terminates at the first non-numeric symbol, such as a blank or the enter key.
- But the enter key and the space can also be a number.

Scanf

- One method is to use ***strings***, but that will not be covered until Chapter 5.
- Another method is to use the following statement

```
fflush(stdin);
```

- After a scanf statement is called.

Scanf

- For example

```
scanf("%c",&MyChar);
fflush(stdin);
```

- will read a single character from the keyboard, and remove any “hit enter” that is at the end of the input.
- This can also be used with %d for integers and %f for floats

User Wait

- Many times, perhaps, it would be best if the program would halt until it is ready to end.
- This waiting can be readily done by using while loops (chapter 2) combined with various input commands.

User Wait

- One simple version of user wait is

```
while('q'!=getchar());
```

- Which will wait until the lower case q is entered on the keyboard. This requires #include<stdio.h>

User Wait

- A second form of user wait requires

```
#include <conio.h>
```

- And looks something like this

```
printf("Press any key to Continue\n");
while(!_kbhit());
_getch();
```

EOF

- EOF is a keyword in C, representing “End of File”, which is a -1.
- On the keyboard, the EOF would be the <CTRL> key followed by the <Z> key.

Simple Menu Example

```
void SimpleMenu()
{
  int MenuFlag=1;
  char KeyStroke='\0';
```

Simple Menu Example

```
while (MenuFlag)
{
   printf("1. Item 1\n");
   printf("2. Item 2\n");
   printf("3. Item 3\n");
   printf("q. Quit\n");
```

Simple Menu Example

```
scanf("%c",&KeyStroke);
fflush(stdin);
```

Simple Menu Example

```
if(KeyStroke=='1')
  printf("Option 1\n");
else if (KeyStroke=='2')
  printf("Option 2\n");
else if (KeyStroke=='3')
  printf("Option 3\n");
else if (KeyStroke=='q')
  MenuFlag=0;
//While Loop will end
//after this line executes.
else
  printf("Unrecoginized Keystroke\n");
```

Simple Menu Example

```
  }//End of While Loop
}//End of Function
```

Why do we use a loop?

- A loop is used whenever a task has to be repeated multiple times.

What is a loop?

- A loop is a special statement in C for controlling a section of code.
- **for** loops and **while** loops

How do we create a loop?

- A while loop is the simplest form of a loop.
- A while loop repeats the following statement or statement block so long as its expression is true
- A while loop requires a condition for the expression

```
int WhileFlag=1;
char Input='0';
printf("Begin While Loop\n");
printf("Enter \'x\' to
   continue\n");
while(WhileFlag)
{
   Input=getchar();
   if(Input=='x')
    WhileFlag=0;
}
  printf("End While Loop\n\n");
}
```

How do we create a loop?

```
int WhileFlag=1;
char Input='0';
printf("Begin While
   Loop\n");
printf("Enter \'x\' to
   continue\n");
while(WhileFlag)
{
   Input=getchar();
   if(Input=='x')
     WhileFlag=0;
}
  printf("End While
   Loop\n\n");
}
```

```
Begin While Loop
Enter 'x' to continue
x
End While Loop
```

How do we create a loop?

- A for loop contains all the conditions of its execution within its use.
- A for loop repeats the following statement or statement block so long as its expression is true.
- A for loop expression will be changed by either the for loop statement, or the following block.

```
int LoopControl=0;
int count=0;
printf("Begin For Loop\n");
for (LoopControl=0;
   LoopControl<10;
   LoopControl++)
{
   printf("LoopControl is
   %d,Count is
   %d\n",LoopControl,count);
   count=count+1;
   LoopControl=LoopControl+1;
   printf("LoopControl is
   %d,Count is
   %d\n",LoopControl,count);
}
printf("Count is %d\n",count);
printf("End For Loop\n\n");
```

How do we create a loop?

```
int LoopControl=0;
int count=0;
printf("Begin For Loop\n");
for (LoopControl=0;
   LoopControl<10;
   LoopControl++)
{
   printf("LoopControl is %d,Count
   is %d\n",LoopControl,count);
   count=count+1;
   LoopControl=LoopControl+1;
   printf("LoopControl is %d,Count
   is %d\n",LoopControl,count);
}
printf("Count is %d\n",count);
printf("End For Loop\n\n");
```

```
Begin For Loop
LoopControl is 0,Count is 0
LoopControl is 1,Count is 1
LoopControl is 2,Count is 1
LoopControl is 3,Count is 2
LoopControl is 4,Count is 2
LoopControl is 5,Count is 3
LoopControl is 6,Count is 3
LoopControl is 7,Count is 4
LoopControl is 8,Count is 4
LoopControl is 9,Count is 5
Count is 5
End For Loop
```

When do we use a loop?

- A key question that needs to be asked is when do you need to use a loop, and what kind?
- If a program contains something that must repeat many times, a loop is used.
- If a program contains something that must repeat many times, and you do not know how many times, a while loop is used.
- If a program contains something that must repeat many times, and you know exactly how many times, then a for loop is used.

When do we use a loop?

- “Input 20 integers and say if they are odd or even”
  - Does it contain a repetition?
  - Yes
  - Is it a fixed size?
  - Yes
  - Does it contain a condition?
  - No
  - What type of loop would be best?
  - **For Loop**

When do we use a loop?

- “Input a series of floats and find the sum of them all until the user says to stop.”
  - Does it contain a repetition?
  - Yes
  - Is it a fixed size?
  - No
  - Does it contain a condition?
  - Yes
  - What type of loop would be best?
  - **While Loop**

When do we use a loop?

- “Input a series of characters and print them out to the screen”
  - Does it contain a repetition?
  - Yes
  - Is it a fixed size?
  - No
  - Does it contain a condition?
  - No
  - What type of loop would be best?
  - **Neither, While Loop or For Loop would work, but a while loop may be a slightly better choice**

What is a Program

- A Program is a series of instructions entered into a computer system to perform a task.

What is a Function?

- A function is a program within a program, that should perform some sub-task of the program.

Prototypes, Function Declaration, Parameters, Return Types

- <return type> function Identifier (<Parameter1, parameter2,...parameter n);
- The return type is a valid data type, for example an int, char, or float.
- The parameters are variable declarations, for example int x, char y, and float z.
- For a prototype, a semicolon is added to the end of the function header.
- An empty parameter list is either blank or of type void.

Calling a Function

```
float Halve(float p)
{
    float result=0.0;
    result=p*0.5;
    return result;
}
```

```
void main()
{
   float value=0.0;
   float half=0.0;
   value=5.0;
   half=Halve(value);
   printf("%f\n",half);
}
```

Behavior of the calling of a function