C Programming

Week 2

Variables, flow control and the Debugger

Example 1: ‘scanf’ use

```c
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Example 1: Summing user variables

```c
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

We read the first variable from the user (5)

We read the second value from the user (7)

Output:
Answer: 5 + 7 = 12

Example 1: Summing user variables

```
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Memory

a	b	sum
???	???	???

Example 1: Summing user variables

```c
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Memory

a	b	sum
???	???	???

Example 1: Summing user variables

```
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Memory

a	b	sum
5	???	???

Example 1: Summing user variables

```c
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Memory

a	b	sum
5	**7**	**???**

Example 1: Summing user variables

```
#include <stdio.h>

int main()
{
   int a, b, sum;

   printf("Enter an integer\n");
   scanf("%d", &a);

   printf("Enter an integer\n");
   scanf("%d", &b);

   sum = a + b;

   printf("Answer: %d + %d = %d\n", a, b, sum);
   return 0;
}
```

Memory

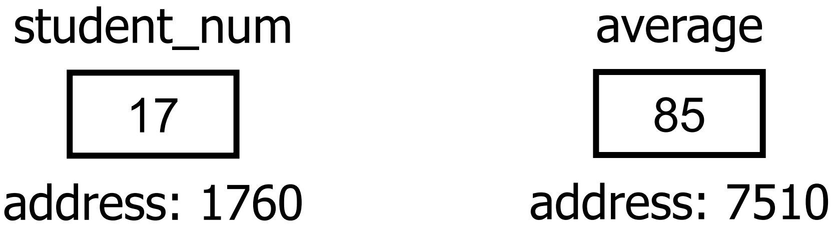
a	b	sum
5	7	12

Why we need & in scanf?

- The & sign refers to the memory address of the variable – where it should be stored.

```
scanf( "%d %lf", student_num, average);
```

Error!!
Will store in location
17 and 5



student_num

17

address: 1760

average

85

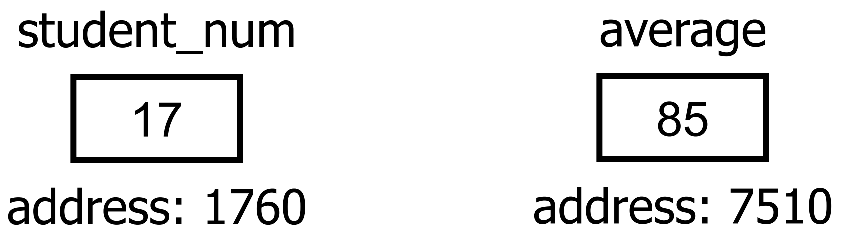
address: 7510

Why we need & in scanf?

- The & sign refers to the memory address of the variable – where it should be stored.

```
scanf( "%d %lf", &student_num, &average);
```

OK!!
Will store in location 1760 and 7510



student_num

17

address: 1760

average

85

address: 7510

scanf

- When reading numbers:
  - The computer expects white space between numbers
    - White space: space, tab, new line.
  - scnaf ignores the white space.

`scanf("%d%d",&i,&j);`	Numbers separated with white space	12 34
`scanf("%d %d",&i,&j);`	Numbers separated with white space	12 34
`scanf("%d, %d",&i,&j);`	Numbers separated with “,”	12,34
`scanf("%d + %d",&i,&j);`	Numbers separated with “+”	12 + 34

scanf

- When reading characters:
  - White spaces are considered as part of the input.

`scanf("%c%c",&i,&j);`	Consecutive characters	ab
`scanf("%c %c",&i,&j);`	Characters separated with space	a b

Shorter lines

`e++;` ⇔ `e = e + 1;`

- We first perform the mathematical operation with the variable (=) and then evaluate it (++).

```
i = 5;
j = i++;
```

→

```
j=5
i=6
```

`++f;` ⇔ `f = f + 1;`

- We first evaluate the variable (++) and then perform the mathematical operation with it (=).

```
i = 5;
j = ++i;
```

→

```
i=j=6
```

Shorter lines

`e--;` ⇔ `e = e - 1;`

- We first perform the mathematical operation with the variable (=) and then evaluate it (--).

```
i = 5;
j = i--;
```

→

```
j=5
i=4
```

`--f;` ⇔ `f = f - 1;`

- We first evaluate the variable (--) and then perform the mathematical operation with it (=).

```
i = 5;
j = --i;
```

→

```
i=j=4
```

Shorter lines

```
e--;
```

⇔

```
e = e - 1;
```

- We first perform the mathematical operation with the variable and then evaluate it.

```
i = 5;
j = i--;
```

→

```
j=5
i=4
```

```
--f;
```

⇔

```
f = f - 1;
```

- We first evaluate the variable and then perform the mathematical operation with it

```
i = 5;
j = --i;
```

→

```
i=j=4
```

Exercise

- Write a program that gets from the user the amount of US $, the current exchange rate and converts it to NIS.

FEDERAL RESERVE NOTE
THE UNITED STATES OF AMERICA
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
L 11180916 G
WASHINGTON, D.C.
ONE DOLLAR
1
שקל חדש

Solution

Exercise_Exchange.c

Understanding Errors

Error:

error LNK2019: **unresolved external symbol _WinMain@16 referenced in function _WinMainCRTStartup**

Problem:

You did not define your project as a console application

Solution:

Start over, create a new project, this time remember to define it as a console application

Understanding Errors

Error:

fatal error C1083: **Cannot open include file: 'stdioh': No such file or directory**

Problem:

The file you’re trying to include does not exist

Solution:

Make sure the file name is spelled correctly

Understanding Errors

Error:

warning C4013: **'print' undefined; assuming extern returning int**

Problem:

The command you’re trying to use does not exist

Solution:

Make sure you spelled the command’s name correctly (upper/lower case, omitting letters etc.)

Understanding Errors

Error:

error C2146: **syntax error : missing ';' before identifier 'print'**

Problem:

Hmm ..., missing ‘;’?

Solution:

What are you waiting for, add it!

scanf warning

- In visual 2008 when you use scanf you might get the following warning:

  "warning C4996: 'scanf' was declared deprecated …"

- You can disregard the warning.

The Debugger

- Some programs may compile correctly, yet not produce the desirable results
- These programs are **valid** and **correct** C programs, yet not the programs we meant to write!
- The debugger can be used to follow the program step by step and may help detecting bugs in an **already compiled** program

Example (for debugging)

- A program that sums the digits of a number with three digits.
- For example:
  - The input 369 yields the output 18

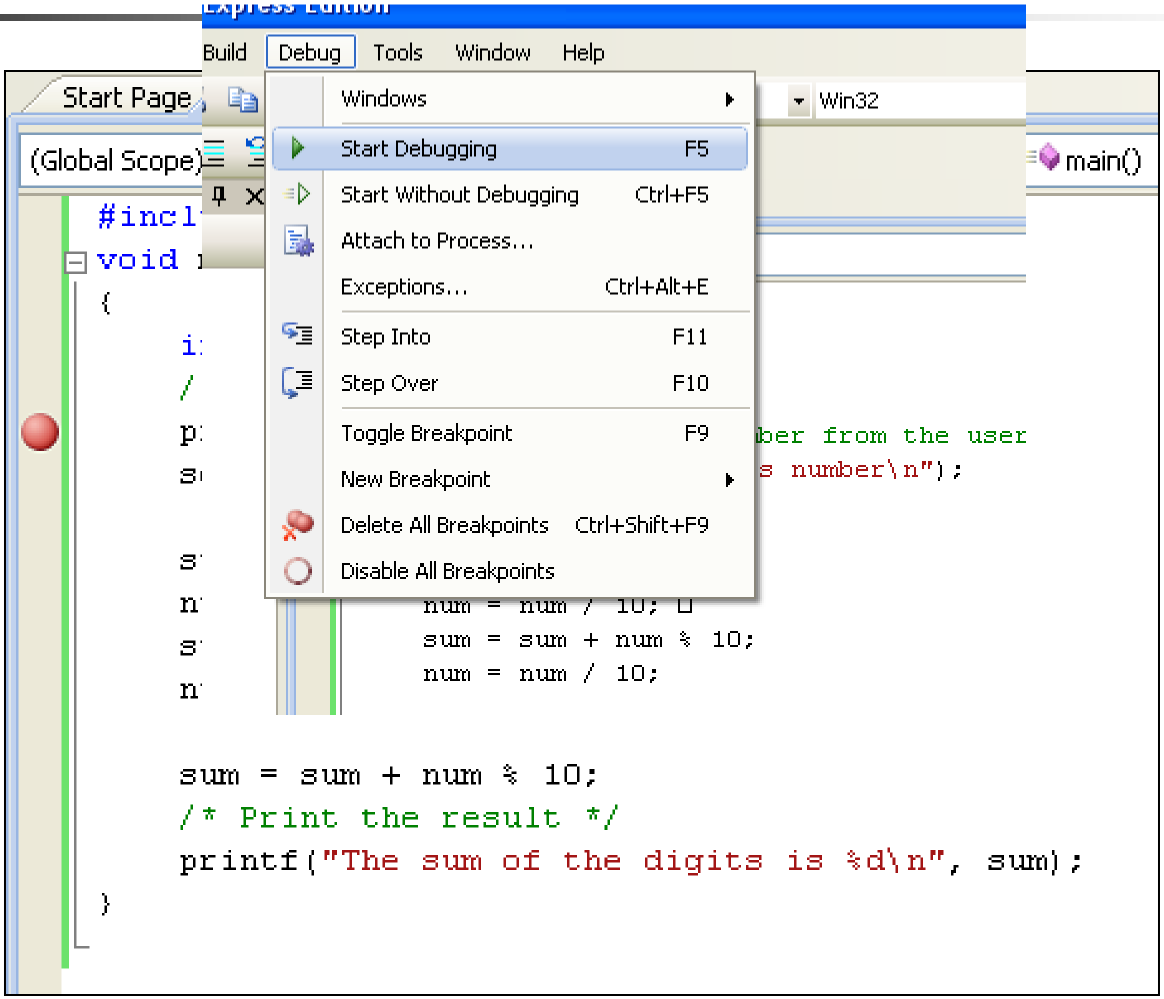
Example 2: Sum Digits

```
#include <stdio.h>
int main()
{
    int sum = 0, num;
    /* Read a 3-digit number from the user */
    printf("Enter 3-digits number\n");
    scanf("%d", &num);

    sum = sum + num % 10;
    num = num / 10;

    sum = sum + num % 10;
    num = num / 10;

    sum = sum + num % 10;
    /* Print the result */
    printf("The sum of the digits is %d\n", sum);
    return 0;
}
```

Exercise

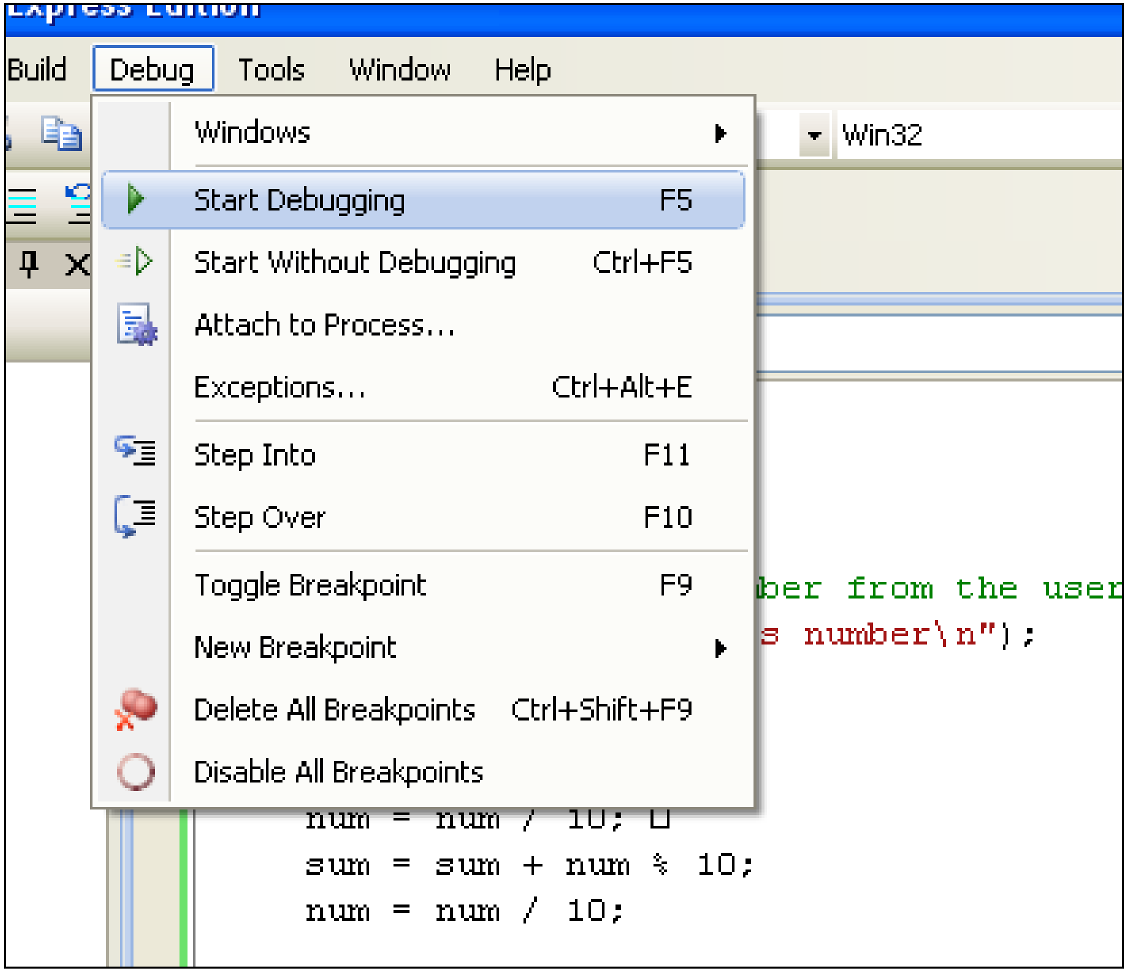
- Copy the above program
- Run in the debugger and see how it works

Insert Breakpoint



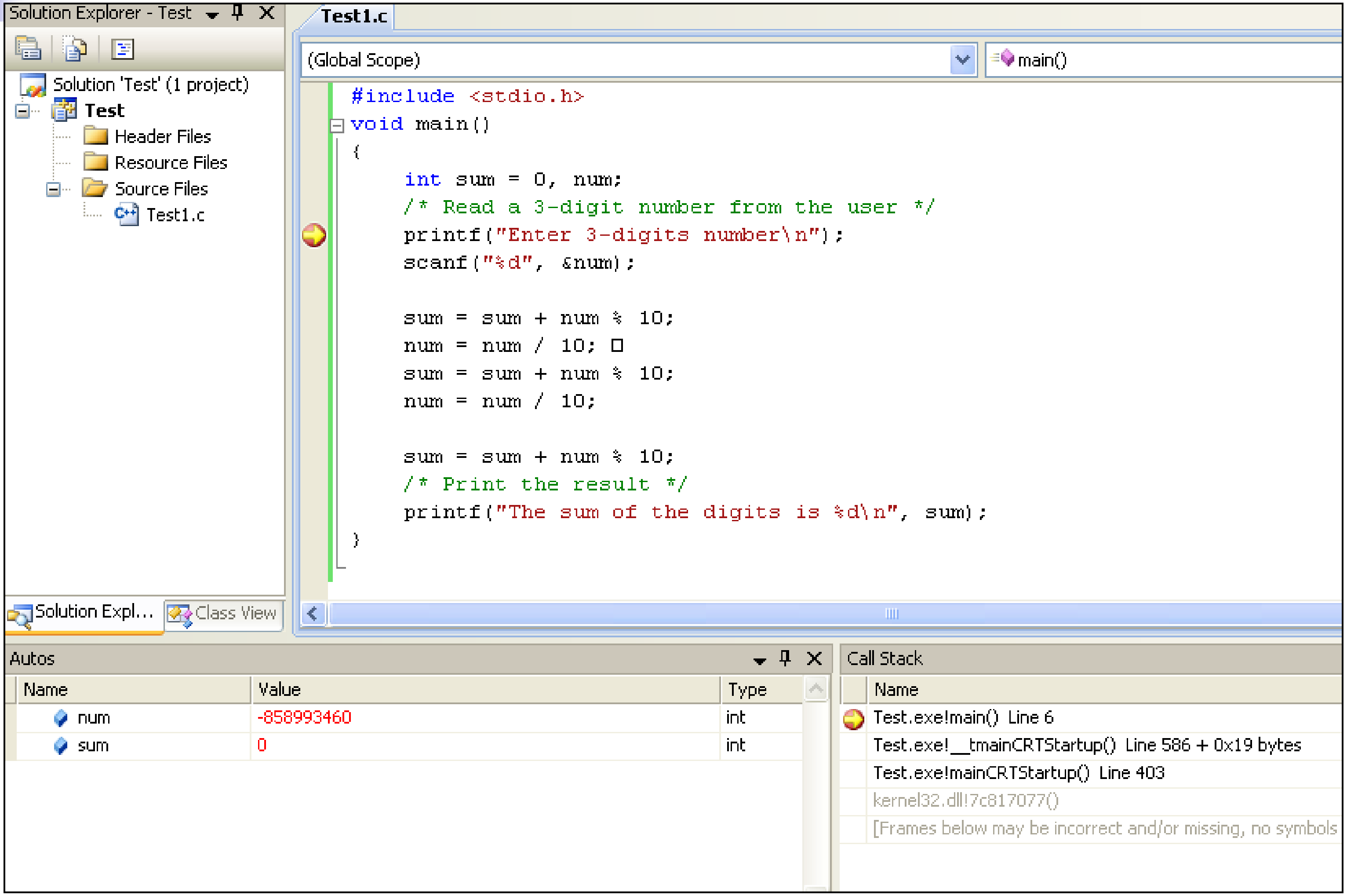
```
#incl
void 
{
    i
    /
    p                                   ber from the user
    s                                   s number\n");

    s
    n                  num = num / 10;
    s                  sum = sum + num % 10;
    n                  num = num / 10;

    sum = sum + num % 10;
    /* Print the result */
    printf("The sum of the digits is %d\n", sum);
}
```

Start Debugging (F5)



```
ber from the user
s number\n");
    num = num / 10; □
    sum = sum + num % 10;
    num = num / 10;
```

Debugger



```
#include <stdio.h>
void main()
{
      int sum = 0, num;
      /* Read a 3-digit number from the user */
      printf("Enter 3-digits number\n");
      scanf("%d", &num);

      sum = sum + num % 10;
      num = num / 10; □
      sum = sum + num % 10;
      num = num / 10;

      sum = sum + num % 10;
      /* Print the result */
      printf("The sum of the digits is %d\n", sum);
}
```

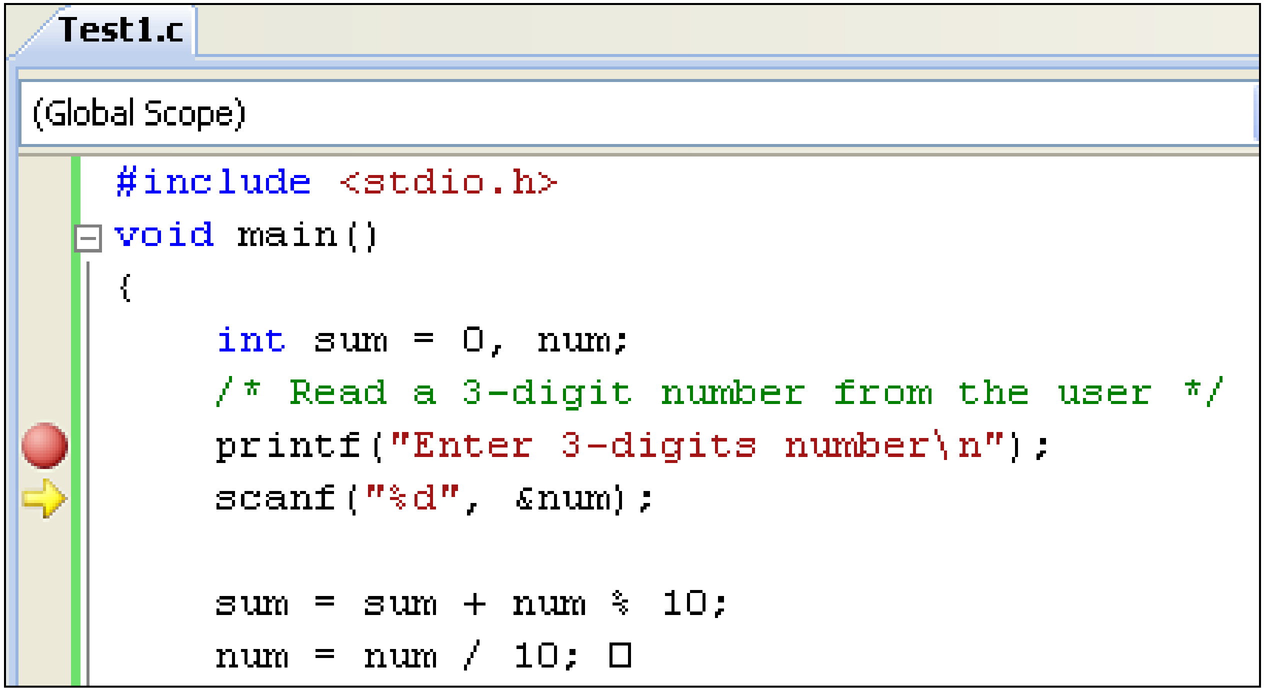
Autos

Name	Value	Type
num	-858993460	int
sum	0	int

Call Stack

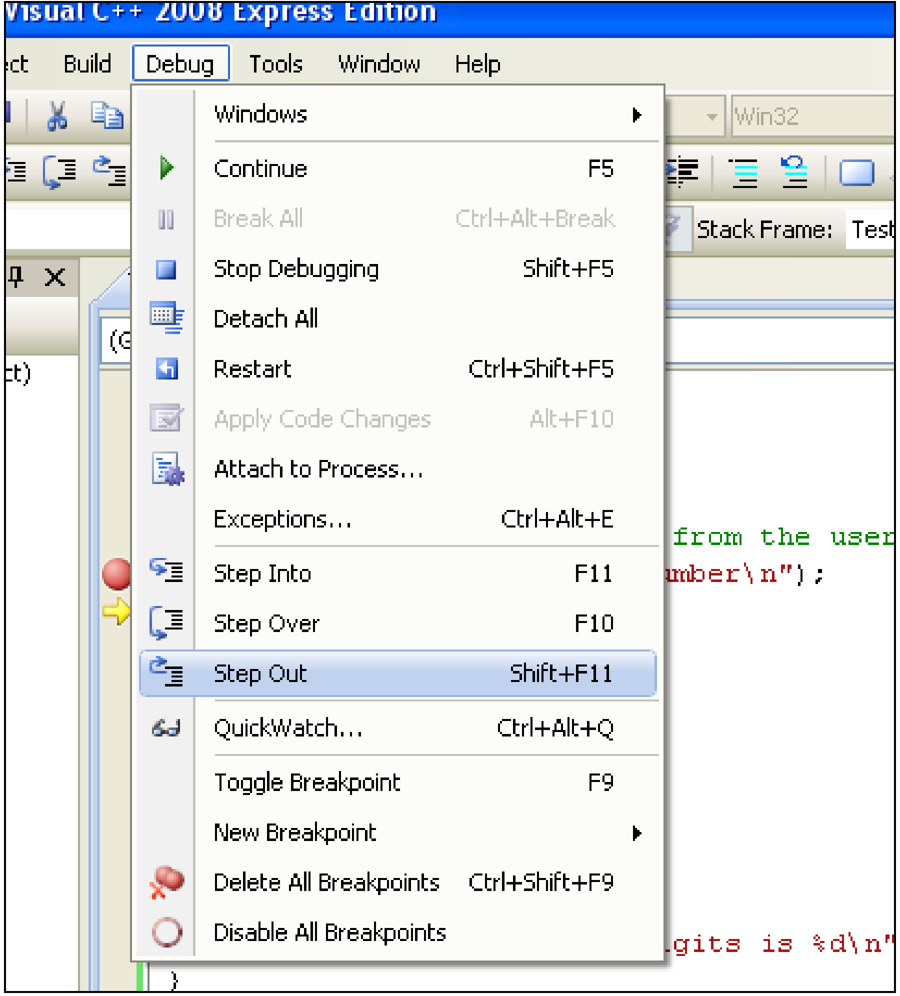
Name
Test.exe!main() Line 6
Test.exe!__tmainCRTStartup() Line 586 + 0x19 bytes
Test.exe!mainCRTStartup() Line 403
kernel32.dll!7c817077()
[Frames below may be incorrect and/or missing, no symbols

Step Over (F10)

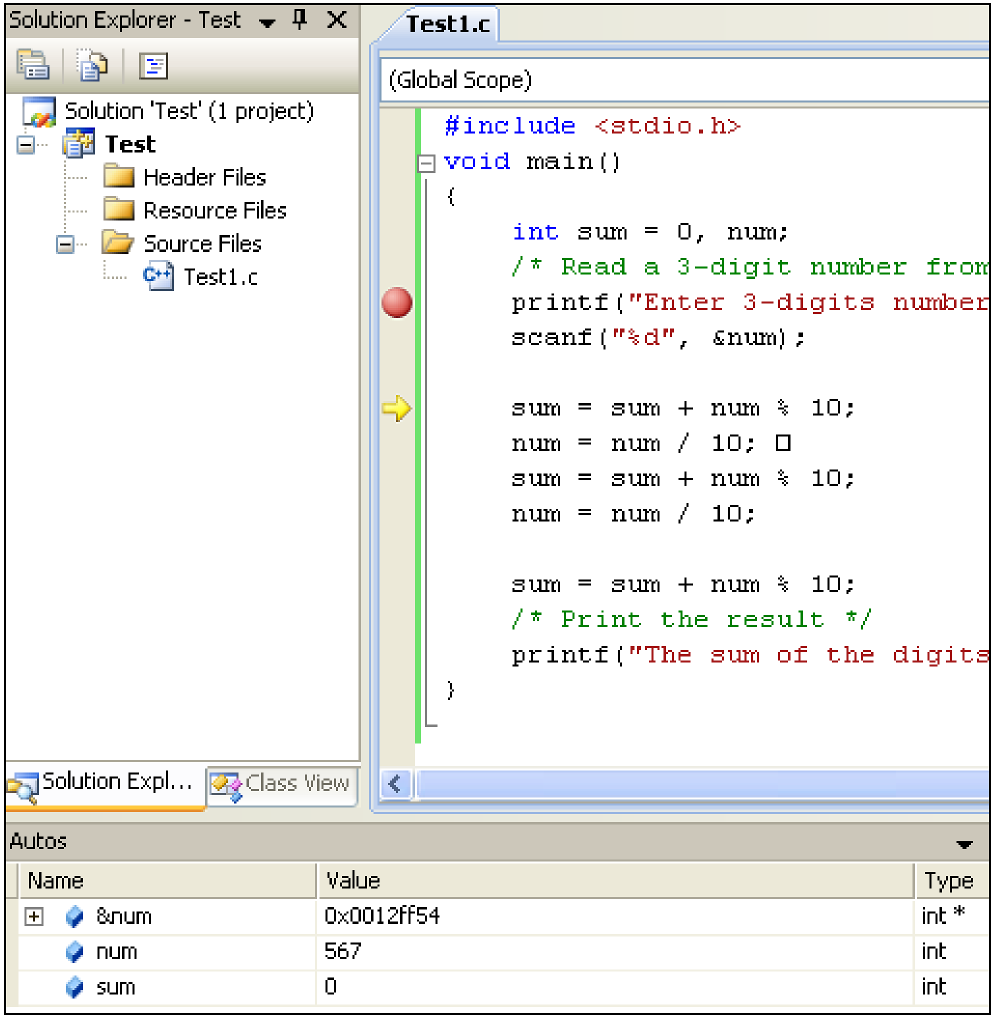


```
#include <stdio.h>
void main()
{
	int sum = 0, num;
	/* Read a 3-digit number from the user */
	printf("Enter 3-digits number\n");
	scanf("%d", &num);

	sum = sum + num % 10;
	num = num / 10; □
```

Other Commands



Update Variables

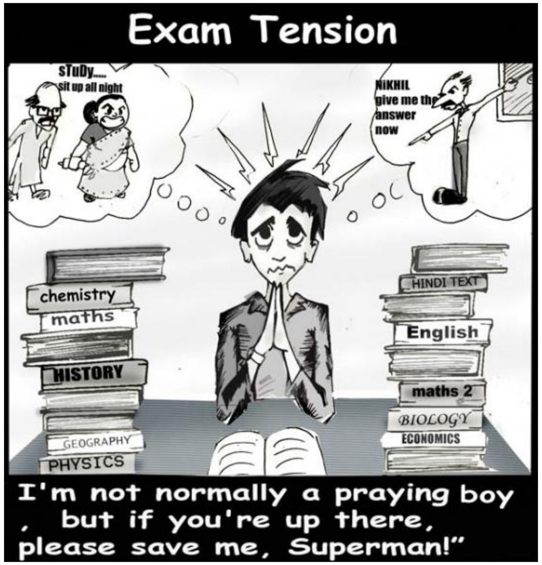


```
#include <stdio.h>
void main()
{
     int sum = 0, num;
     /* Read a 3-digit number from
     printf("Enter 3-digits number
     scanf("%d", &num);

     sum = sum + num % 10;
     num = num / 10; □
     sum = sum + num % 10;
     num = num / 10;

     sum = sum + num % 10;
     /* Print the result */
     printf("The sum of the digits
}
```

Name	Value	Type
⊞ &num	0x0012ff54	int *
num	567	int
sum	0	int

Example 3

- Write a program that gets from the user a number and returns its absolute value



Example 3 - Absolute Value

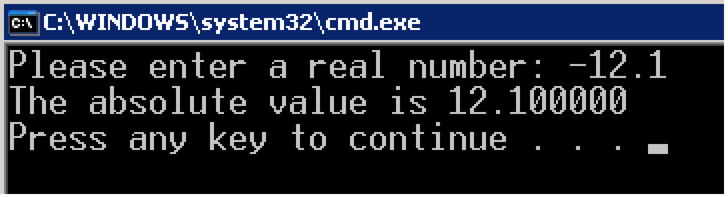
```
/* The absolute value of a number */
int main()
{
  double num = 0.0;

  printf("Please enter a real number: ");
  scanf("%lf", &num);

  if (num < 0)
      num = -num;

  printf("The absolute value is %lf\n", num);
  return 0;
}
```

Example 3 - Absolute Value

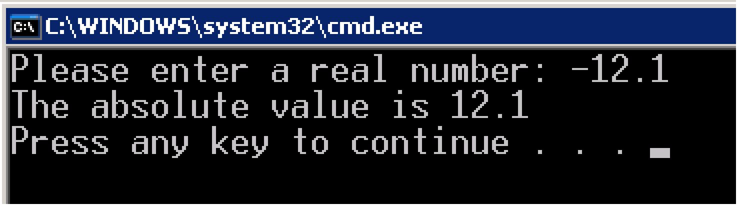
```
/* The absolute value of a number */
int main()
{
  double num = 0.0;

  printf("Please enter a real number: ");
  scanf("%lf", &num);

  if (num < 0)
      num = -num;

  printf("The absolute value is %g\n", num);
  return 0;
}
```

What happens if we change %lf to %g?

From %lf to %g



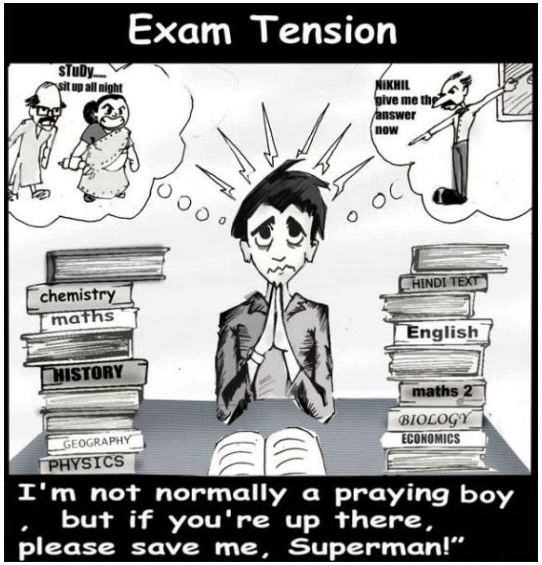
```
Please enter a real number: -12.1
The absolute value is 12.100000
Press any key to continue . . .
```



```
Please enter a real number: -12.1
The absolute value is 12.1
Press any key to continue . . .
```

Example 4

- Write a program that gets from the user a grade of a student in an exam and prints the grade, adding if the student passed or failed the exam.
  - If the grade is 60 or above – the student passed the exam.



Example 4 – student grade

```
// The grade of a student - did he pass the exam?
int main()
{
   int grade = 0;

   printf("Please enter a the grade of the student: ");
   scanf("%d", &grade);
   if (grade >=60)
       printf ("The grade is %d, the student passed the
   exam\n",grade);
   else
       printf ("The grade is %d, the student failed the
   exam\n",grade);
   return 0;
}
```

Example 5

- Write a program that is given a 0-100 grade and translates it into A-F grade
- Conversion:
  - 90-100: A
  - 80-89: B
  - 70-79: C
  - 60-69: D
  - 0-59: F

Example 5 – grade conversion

```
// Converting a grade from 0-100 to A-F
int main()
{
   int grade = 0;

   printf("Please enter a the grade of the student: ");
   scanf("%d", &grade);
   printf ("The grade is ");
   if (grade >= 90) {
    printf ("A\n");
   } else if (grade >= 80) {
        printf ("B\n");
   } else if (grade >= 70) {
        printf ("C\n");
   } else if (grade >= 60) {
        printf ("D\n");
   } else {
        printf ("F\n"); }

   return 0;
}
```