



2. Biến

- Biến là một vùng nhớ được lưu tại một địa chỉ nào đó trong bộ nhớ máy tính.
- Biến được đặt tên thông qua khai báo biến
- Giá trị mà biến lưu trữ có thể bị thay đổi nhiều lần trong suốt quá trình chương trình thi hành.



Chương 2: Các kiểu dữ liệu cơ sở

Cú pháp:

<kiểu> <Tên biến>;

<kiểu> <Tên biến 1>, <Tên biến 2>;

Ví dụ:

int i;

int j, k;

unsigned char dem;

float ketqua, delta;



3. Hằng

- Hằng cũng là một vùng nhớ được lưu trữ tại một địa chỉ nào đó trong bộ nhớ máy tính .
- Hằng được đặt tên thông qua khai báo hằng.
- Giá trị mà hằng lưu trữ **không thay đổi** trong suốt quá trình chương trình thi hành.



Cú pháp:

lưu ý ở cách 1 ko có dấu ;

cách1 **#define** <Tên hằng> <Giá trị> *không có quy định kiểu dữ liệu*

cách2 **const** <kiểu> <Tên hằng> = <Giá trị>; *có quy định kiểu dữ liệu*

Ví dụ:

#define MAX 100 *vì ở đây không có kiểu, nên khi printf ra máy sẽ tự hiểu*

#define PI 3.14

Hoặc

const int MAX = 100;

const float PI = 3.14;



4. Biểu thức

- BT Là một sự kết hợp giữa các toán tử (+, -, *, /, %....) và các toán hạng (hằng, biến, lời gọi hàm, ...).
- Mỗi BT đều có một giá trị.

Ví dụ:

```
double p = (a + b + c) / 2 ; // Nửa chu vi
```

```
double s = sqrt(p*(p-a)*(p-b)*(p-c)); //Diện tích tam giác
```



a. Toán tử gán

Dùng để gán giá trị cho biến.

Cú pháp:

$\text{<biến> = <biểu thức>;}$

Ví dụ:

```
int a, b, c, d, e;
```

```
a = 12;
```

```
b = a;
```

```
c = a + b;
```

```
e = d = c; //Phép gán liên tiếp
```



b. Toán tử số học

- Toán tử 1 ngôi: Là toán tử chỉ tác động lên một toán hạng
++ (tăng 1 đơn vị) -- (giảm 1 đơn vị)

Chú ý:

Khi toán tử đặt trước toán hạng thì **toán hạng được tăng/giảm trước khi được dùng.**

còn khi toán tử đặt sau toán hạng thì **toán hạng được dùng trước khi tăng/giảm.**



Chương 2: Các kiểu dữ liệu cơ sở

Ví dụ:

```
int x, y;
```

```
x = 10;
```

```
y = x++ + 5; // y = 15 và x = 11
```

x sẽ = x+1 rồi sau đó +5

```
x = 10;
```

```
y = ++x + 5; // x = 11 và y = 16
```

x sẽ = x+1 trước khi vào rồi sau đó mới +5



Chương 2: Các kiểu dữ liệu cơ sở



- Toán tử 2 ngôi: Là toán tử tác động lên hai toán hạng.

+ - * / % (chia lấy phần dư) += -= *=
/= %=

Ví dụ 1:

```
int a = 9, b = 2, x, y;
```

```
float c = 9.0, z;
```

```
x = a/b; // x = 4 (chia nguyên)
```

```
y = a%b; // y = 1
```

```
z = c/b; // z = 4.5 (chia thực)
```



Ví dụ 2:

int n;

n = 10;

n += 10; // $n = n + 10 \rightarrow n = 20$

n -= 5; // $n = n - 5 \rightarrow n = 15$

n *= 2; // $n = n * 2 \rightarrow n = 30$

n /= 4; // $n = n / 4 \rightarrow n = 7$

n %= 2; // $n = n \% 2 \rightarrow n = 1$



c. Toán tử quan hệ

Toán tử quan hệ cho ra giá trị đúng hoặc sai.

`==` (bằng) `>` `<` `>=` `<=` `!=` (khác)

Ví dụ:

`bool b1 = (1 == 2);` `//b1 → false`

`bool b2 = (1 != 2);` `//b2 → true`

`bool b3 = (1 > 2);` `//b3 → false`

`bool b4 = (1 >= 2);` `//b4 → false`

`bool b5 = (1 < 2);` `//b5 → true`

`bool b6 = (1 <= 2);` `//b6 → true`



d. Toán tử luận lý

Toán tử luận lý cho ra giá trị đúng/sai. && (and) || (or) ! (not)

<u>a</u>	<u>b</u>	<u>a && b</u>	<u>a b</u>	<u>!a</u>
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Ví dụ:

```
bool b1 = (1 > 2) && (3 > 4); //b1 → false
```

```
bool b2 = (1 < 2) || (3 > 4); //b2 → true
```

```
bool b3 = !b2; //b3 → false
```



Chương 2: Các kiểu dữ liệu cơ sở

**e. Toán tử biểu thức điều kiện**

Đây là toán tử 3 ngôi (gồm có 3 toán hạng), ký hiệu ?:

Cú pháp:

$\langle \text{Bt1} \rangle ? \langle \text{Bt 2} \rangle : \langle \text{Bt 3} \rangle$

biểu thức này cho kết quả là $\langle \text{Bt2} \rangle$ nếu $\langle \text{Bt1} \rangle$ đúng, ngược lại cho kết quả là $\langle \text{Bt3} \rangle$



Chương 2: Các kiểu dữ liệu cơ sở

Ví dụ:

```
int a = 10, b = 20, max, min;
```

```
max = (a > b) ? a : b;
```

```
min = (a < b) ? a : b;
```



f. Toán tử chuyển kiểu

Toán tử chuyển kiểu dùng để chuyển một kiểu dữ liệu bất kỳ sang một kiểu dữ liệu mong muốn.

Cú pháp: (<Kiểu>) <Biểu>

Ví dụ:

```
int a = 9, b = 2;
```

```
int c = a / b; //4
```

```
float d = (float)a / b; //4.5
```

chỉ chuyển kiểu của a xong rồi chia



5. Các hàm thư viện C chuẩn

- Hàm thư viện là những hàm đã được định nghĩa sẵn trong một thư viện nào đó của C.
- Muốn sử dụng các hàm thư viện thì phải khai báo thư viện trước khi sử dụng bằng chỉ thị **#include <tên thư viện.h>** đặt ở đầu tập tin chương trình.



Chương 2: Các kiểu dữ liệu cơ sở



a. Hàm toán học

Các hàm toán học được định nghĩa sẵn trong thư viện **math.h**.

**Chương 2: Các kiểu dữ liệu cơ sở**

Tên hàm	Trả về
ceil(x)	Làm tròn lên của x
floor(x)	Làm tròn xuống của x
abs(x)	Trị tuyệt đối của x nguyên
fabs(x)	Trị tuyệt đối của x thực
sqrt(x)	Căn bậc hai của x
pow(x, y)	x lũy thừa y
exp(x)	exponential của x
log(x)	Logarithm tự nhiên của x



Chương 2: Các kiểu dữ liệu cơ sở

Ví dụ:

```
float x = 3.7;
```

```
int y = ceil(x); //y = 4
```

```
int z = floor(x); //z = 3
```

```
int a = -10; int b = abs(a); //b = 10;
```

```
float u = -20.5; float v = fabs(u); //v = 20.5
```

```
float t = pow(2.0, 10.0); // t =  $2^{10} = 1024$ 
```

```
float p = exp(log(100.0)/3); //y =  $e^{(1/3)\log(100)} = 100^{1/3}$ 
```

```
float q = exp(10.0*log(3.0)); //q =  $e^{10\log(3)} = 3^{10}$ 
```



Chương 2: Các kiểu dữ liệu cơ sở

**b. Hàm xuất dữ liệu**

Hàm xuất dữ liệu được định nghĩa sẵn trong thư viện `stdio.h`.

Cú pháp:

```
printf(<chuỗi định dạng>[, <ts1>, <ts2>,  
...]);
```



Chương 2: Các kiểu dữ liệu cơ sở



- <chuỗi định dạng> là cách trình bày thông tin xuất và được đặt trong cặp nháy kép “”, gồm 3 loại:
 - Văn bản thường (literal text): Được xuất y hệt như lúc gõ trong chuỗi định dạng.
 - Ký tự điều khiển (escape sequence): Gồm dấu \ và một trong các ký tự như trong bảng sau:



Chương 2: Các kiểu dữ liệu cơ sở

Ký tự điều khiển

\a

\n

\t

\\

\”

Ý nghĩa

Tiếng chuông

Xuống dòng

Dấu Tab

In dấu \

In dấu “



Chương 2: Các kiểu dữ liệu cơ sở



- Đặc tả (conversion specifier): Gồm dấu % và một vài ký tự, nó xác định kiểu của biến/giá trị muốn xuất.

*%d sử dụng cho
char để cho số
decimal và int*

Đặc tả	Ý nghĩa
%c	Ký tự (char)
%d, %ld	Số nguyên có dấu (char, int, short, long)
%f, %lf	Số thực (float, double)
%s	Chuỗi ký tự (char [], char*)
%u	Số nguyên không dấu (unsigned int/short/long)

**Chương 2: Các kiểu dữ liệu cơ sở**

- Các tham số <ts1>, <ts2>, . . . chính là các biến/giá trị muốn xuất, được liệt kê theo thứ tự cách nhau dấu phẩy

Ví dụ:

```
int a = 10, b = 20;  
printf("%d\n", a); //10  
printf("%d\n", b); //20  
printf("%d %d\n", a, b); //10 20  
float x = 15.06;  
printf("%f\n", x); //15.060000  
printf("%0.2f\n", x); //15.06  
printf("%f\n", 1.0/3); //0.333333  
printf("%0.1f\n", 1.0/3); //0.3
```




c. Hàm nhập dữ liệu

Cú pháp:

scanf(<chuỗi định dạng>[, <đs1>,
<đs1>, ...]);

- <chuỗi định dạng> giống định dạng xuất nhưng chỉ có các đặc tả.
- Các tham số <ts1>, <ts2>, ... là tên các biến sẽ chứa giá trị nhập và được đặt trước bởi toán tử &



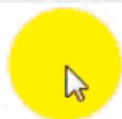
Chương 2: Các kiểu dữ liệu cơ sở

Ví dụ:

```
int a, b;  
scanf("%d", &a); // Nhập giá trị cho a  
scanf("%d", &b); // Nhập giá trị cho b  
scanf("%d%d", &a, &b); // Nhập giá trị cho a và b  
float c, d;  
scanf("%f", &c); // Nhập giá trị cho c  
scanf("%f", &d); // Nhập giá trị cho d  
scanf("%f%f", &c, &d); // Nhập giá trị cho c và d  
char e, f;  
scanf("%c", &e); // Nhập giá trị cho e  
scanf("%c", &f); // Nhập giá trị cho f  
scanf("%c%c", &e, &f); // Nhập giá trị e và f
```

*<Lưu ý> Phải chuột phải project >
properties > C/C++ -> Preprocessor ->
chọn tam giác > Edit -> thêm
CRT_NO_WARNING*

```
Test.cpp -> X
LapTrinhBanBan (Global Scope)
40 //
41 // 9.0 / 2 = 4.5
42 // 9 / 2.0 = 4.5
43 // 9.0 / 2.0 = 4.5
44
45 int a, b;
46
47 printf("\nNhap so nguyen a: ");
48 scanf("%d", a);
49
50 }
51
```



159% 0 3 1 Error 6 Warnings 0 Messages Build + IntelliSense Ln: 51 Ch: 1 TABS CRLF

Error List					
Entire Solution 1 Error 6 Warnings 0 Messages Build + IntelliSense					
Search Error List					
	Code	Description	Project	File	Line
	C6031	Return value ignored: 'scanf'.	LapTrinhBanBan	Test.cpp	48
	C6066	Non-pointer passed as _Param_2 when pointer is required in call to 'scanf' Actual type: 'int'.	LapTrinhBanBan	Test.cpp	48
	C6001	Using uninitialized memory 'a'.	LapTrinhBanBan	Test.cpp	48
	C4326	return type of 'main' should be 'int' instead of 'void'	LapTrinhBanBan	Test.cpp	4
	C4477	'scanf': format string '%d' requires an argument of type 'int *', but variadic argument 1 has type 'int'	LapTrinhBanBan	Test.cpp	48
	C4896	'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.	LapTrinhBanBan	Test.cpp	48
	C4101	'b': unreferenced local variable	LapTrinhBanBan	Test.cpp	45

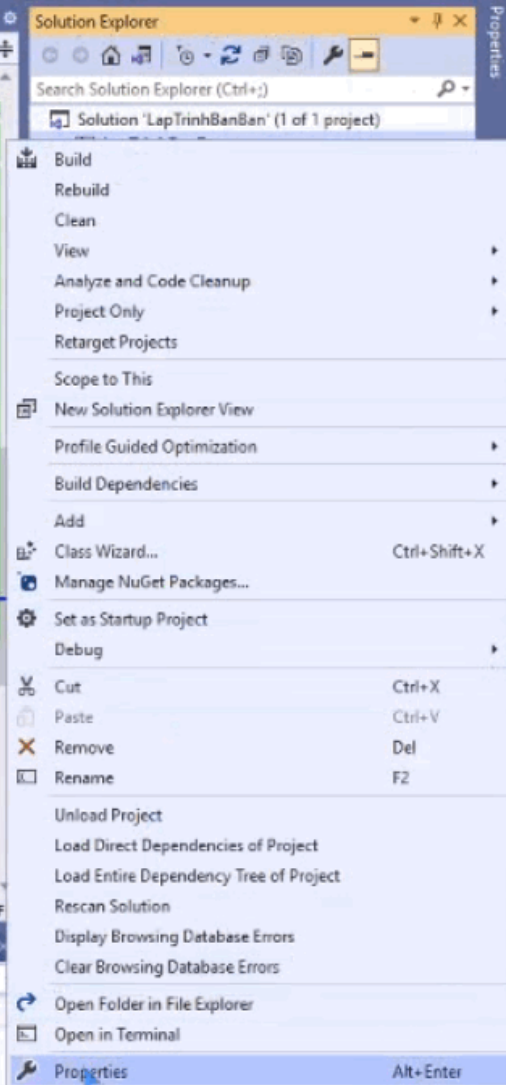
Error List

Entire Solution 1 Error 6 Warnings 0 Messages Build + IntelliSense Search Error List

	Code	Description	Project	File	Line	Suppression State
	C6031	Return value ignored: 'scanf'.	LapTrinhBanBan	Test.cpp	48	
	C6066	Non-pointer passed as _Param_(2) when pointer is required in call to 'scanf' Actual type: 'int'.	LapTrinhBanBan	Test.cpp	48	
	C6001	Using uninitialized memory 'a'.	LapTrinhBanBan	Test.cpp	48	
	C4326	return type of 'main' should be 'int' instead of 'void'	LapTrinhBanBan	Test.cpp	4	
	C4477	'scanf' : format string '%d' requires an argument of type 'int *', but variadic argument 1 has type 'int'	LapTrinhBanBan	Test.cpp	48	
	C4996	'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.	LapTrinhBanBan	Test.cpp	48	
	C4101	'b': unreferenced local variable	LapTrinhBanBan	Test.cpp	45	

*đây là lỗi scanf_s do
version khác biệt*

Lưu ý: 1 solution sẽ có nhiều project, ở đây thì click phải vào project > chọn properties



Configuration: Active(Debug) v

Platform: Active(Win32) v

Configuration Manager...

Configuration Properties

General

Advanced

Debugging

VC++ Directories

C/C++

General

Optimization

Preprocessor

Code Generation

Language

Precompiled Headers

Output Files

Browse Information

External Includes

Advanced

All Options

Command Line

Linker

Manifest Tool

XML Document Generator

Browse Information

Build Events

Custom Build Step

Code Analysis

Preprocessor Definitions

WIN32; DEBUG; _CONSOLE; %(PreprocessorDefinitions)

Undefine Preprocessor Definitions

Undefine All Preprocessor Definitions

No

Ignore Standard Include Paths

No

Preprocess to a File

No

Preprocess Suppress Line Numbers

No

Keep Comments

No

Use Standard Conforming Preprocessor



Preprocessor Definitions

Defines a preprocessing symbols for your source file.

OK

Cancel

Apply

Configuration: Active(Debug) v

Platform: Active(Win32) v

Configuration Manager...

Configuration Properties

General

Advanced

Debugging

VC++ Directories

C/C++

General

Optimization

Preprocessor

Code Generation

Language

Precompiled Headers

Output Files

Browse Information

External Includes

Advanced

All Options

Command Line

Linker

Manifest Tool

XML Document Generator

Browse Information

Build Events

Custom Build Step

Code Analysis

Preprocessor Definitions

WIN32;_DEBUG;_CONSOLE;%(PreprocessorDefinitions) v

Undefine Preprocessor Definitions

<Edit...>

Undefine All Preprocessor Definitions

<inherit from parent or project defaults>

Ignore Standard Include Paths

Preprocess to a File

No

Preprocess Suppress Line Numbers

No

Keep Comments

No

Use Standard Conforming Preprocessor

Preprocessor Definitions

Defines a preprocessing symbols for your source file.

OK

Cancel

Apply

Configuration: Active(Debug)

Platform: Active(Win32)

Configuration Manager...

Configuration Properties

Preprocessor Definitions

WIN32;_DEBUG;_CONSOLE;%(PreprocessorDefinitions)

General

Advanced

Debugging

VC++

C/C++

Preprocessor Definitions

?

X

WIN32
_DEBUG
_CONSOLE
_CRT_SECURE_NO_WARNINGS

 *thêm cái này vào*

Evaluated value:

WIN32
_DEBUG
_CONSOLE
_CRT_SECURE_NO_WARNINGS
%(PreprocessorDefinitions)

Inherited values:

_UNICODE
UNICODE

☒ Inherit from parent or project defaults

Macros >

OK

Cancel

Build + Int

Project

File

Search B

Line

Configuration: Active(Debug) v

Platform: Active(Win32) v

Configuration Manager...

Configuration Properties

General

Advanced

Debugging

VC++ Directories

C/C++

General

Optimization

Preprocessor

Code Generation

Language

Precompiled Headers

Output Files

Browse Information

External Includes

Advanced

All Options

Command Line

Linker

Manifest Tool

XML Document Generator

Browse Information

Build Events

Custom Build Step

Code Analysis

Preprocessor Definitions

WIN32;_DEBUG;_CONSOLE;_CRT_SECURE_NO_WARNINGS;%(Prepro v

Undefine Preprocessor Definitions

Undefine All Preprocessor Definitions

No

Ignore Standard Include Paths

No

Preprocess to a File

No

Preprocess Suppress Line Numbers

No

Keep Comments

No

Use Standard Conforming Preprocessor

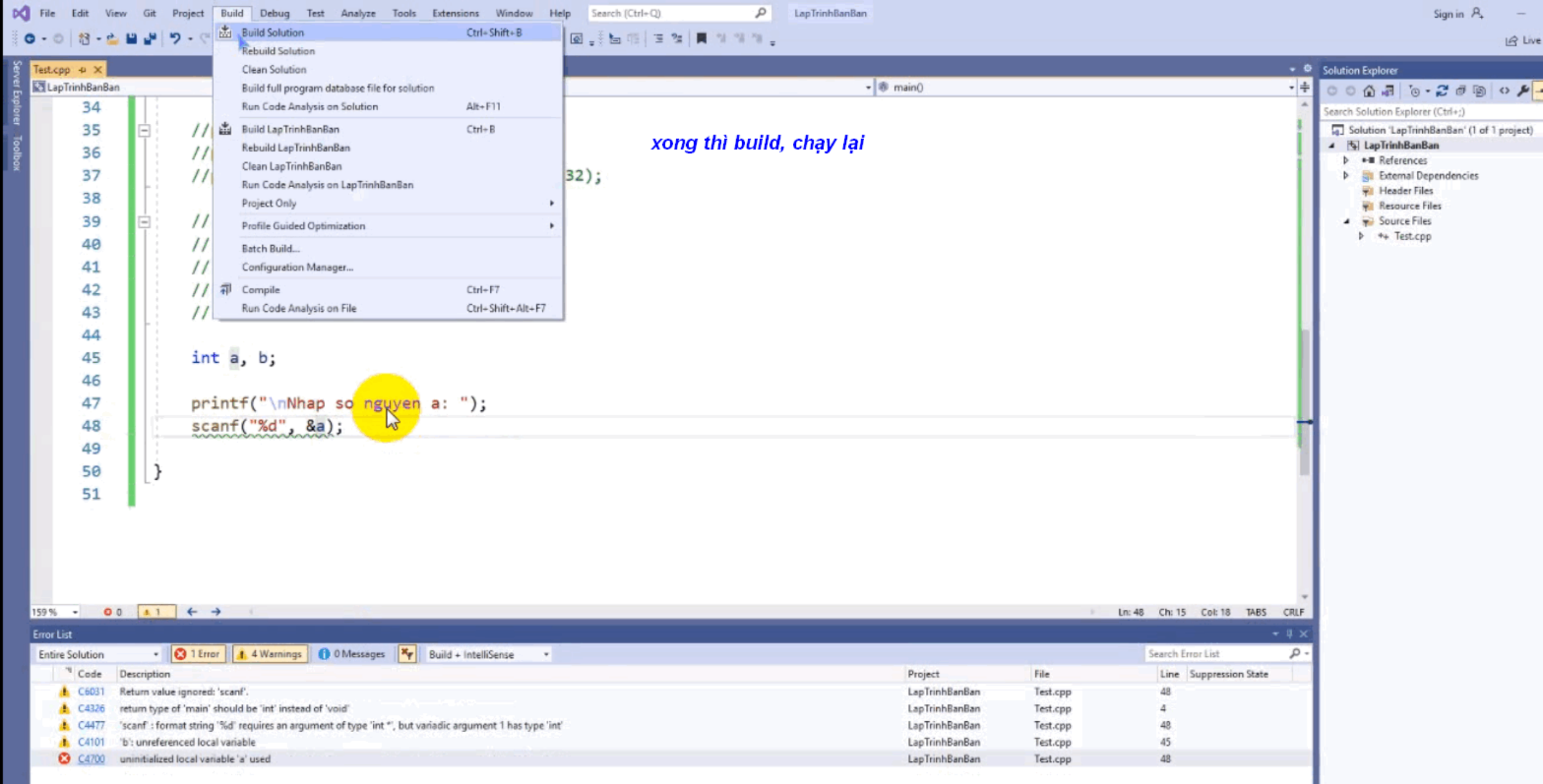
Preprocessor Definitions

Defines a preprocessing symbols for your source file.

OK

Cancel

Apply



```
58     printf("\nNhap so thuc f: ");
59     scanf("%f", &f);
60
61     printf("\nf = %0.2f", f);
62
63     char c;
64
65     printf("\nNhap ky tu c: ");
66     scanf("%c", &c);
67
68     printf("\n\nKy tu: c = %c", c);
69     printf("\nSo nguyen: c = %d", c);
70     printf("\nKy tu in hoa: c = %c", c + 32);
71 }
72
```



Nhap so nguyen a: 10

Nhap so nguyen b: 2

a = 10, b = 2

Tong: $10 + 2 = 12$

Nhap so thuc f: 5.666

f = 5.67

Nhap ky tu c:

Ky tu: c =

So nguyen: c = 10

Ky tu in hoa: c = *

C:\Users\nvdangkhoa\Desktop\LapTrinhBanBan\Debug\LapTrinhBanBan.exe (process 13896) exited with code 0.

Press any key to close this window . . .

*thấy ở đây không thể nhập ký tự ở đây do sau
khi nhập số ở dòng code trên thì phím enter đã
được lưu lại trong bộ nhớ*

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

```
55
56     float f;
57
58     printf("\nNhap so thuc f: ");
59     scanf("%f", &f);
60
61     printf("\nf = %0.2f", f);*/
62
63     char c;
64
65     printf("\nNhap ky tu c: ");
66     scanf("%c", &c);
67
68     printf("\n\nKy tu: c = %c", c);
69     printf("\nSo nguyen: c = %d", c);
70     printf("\nKy tu in thuong: c = %c", c + 32);
71 }
72
```

Nhap ky tu c: A

trường hợp này thì bình thường

Ky tu: c = A

So nguyen: c = 65

Ky tu in thuong: c = a

C:\Users\nvdangkhoa\Desktop\LapTrinhBanBan\Debug\LapTrinhBanBan.exe (process 23516) exited with code 0.

Press any key to close this window . . .

```
52
53     printf("\na = %d, b = %d", a, b);
54     printf("\nTong: %d + %d = %d", a, b, a + b);
55
56     float f;
57
58     printf("\nNhap so thuc f: ");
59     scanf("%f", &f);
60
61     printf("\nf = %0.2f", f);
62
63     char c;
64
65     printf("\nNhap ky tu c: ");
66     // fflush(stdin);
67     rewind(stdin);
68     scanf("%c", &c);
69
70     printf("\n\nKy tu: c = %c", c);
71     printf("\nSo nguyen: c = %d", c);
72     printf("\nKy tu in thuong: c = %c", c + 32);
73 }
```

*sử dụng hàm này để
giải phóng bộ nhớ*

Nhap so nguyen a: 10

Nhap so nguyen b: 2

a = 10, b = 2

Tong: $10 + 2 = 12$

Nhap so thuc f: 5.6

f = 5.60

Nhap ky tu c: A **đã ngừng ở đây để
đợi nhập ký tự vào**



Nhap so nguyen a: 10

Nhap so nguyen b: 2

a = 10, b = 2

Tong: $10 + 2 = 12$

Nhap so thuc f: 5.6

f = 5.60

Nhap ky tu c: A

Ky tu: c = A

So nguyen: c = 65

Ky tu in thuong: c = a

C:\Users\nvdangkhoa\Desktop\LapTrinhBanBan\Debug\LapTrinhBanBan.exe (process 9016) exited with code 0.

Press any key to close this window . . .

```
1  #include <stdio.h>
```

```
2
```

```
3  #include <math.h>
```

sử dụng thư viện math.h

```
4
```

```
5  void main()
```

```
6  {
```

```
7
```

```
8
```

```
    //printf("Chao cac ban"); /*Xuất dữ liệu ra màn hình*/
```

```
    //printf("\nTrung tam tin hoc");
```

```
67 //printf("\nNhap ky tu c: ");
68 //// fflush(stdin);
69 //rewind(stdin);
70 //scanf("%c", &c);
71
72 //printf("\n\nKy tu: c = %c", c);
73 //printf("\nSo nguyen: c = %d", c);
74 //printf("\nKy tu in thuong: c = %c", c + 32);
75
76 int n = 8;
77
78 sqrt(),
79 }
80
```

hàm sẽ trả về kiểu double

phải làm đúng format của hàm

hàm cần biến kiểu double

```
67 //printf("\nNhap ky tu c: ");
68 //// fflush(stdin);
69 //rewind(stdin);
70 //scanf("%c", &c);
71
72 //printf("\n\nKy tu: c = %c", c);
73 //printf("\nSo nguyen: c = %d", c);
74 //printf("\nKy tu in thuong: c = %c", c + 32);
75
76 int n = 8;
77
78 double kq1 = sqrt((double)n);
79 }
80
```

```
70 //scanf("%c", &c);
71
72 //printf("\n\nKy tu: c = %c", c);
73 //printf("\nSo nguyen: c = %d", c);
74 //printf("\nKy tu in thuong: c = %c", c + 32);
75
76 int n = 8;
77
78 double kq1 = sqrt((double)n);
79
80 printf("\nkq1 = %lf", kq1);
81 }
```

kq1 = 2.828427

C:\Users\nvdangkhoa\Desktop\LapTrinhBanBan\Debug\LapTrinh

Press any key to close this window . . .

```
70 //scanf("%c", &c);
71
72 //printf("\n\nKy tu: c = %c", c);
73 //printf("\nSo nguyen: c = %d", c);
74 //printf("\nKy tu in thuong: c = %c", c + 32);
75
76 int n = 8;
77
78 double kq1 = sqrt((double)n);
79 float kq2 = (float) sqrt((double)n);
80 int kq3 = (int) sqrt((double)n);
81
82 printf("\nkq1 = %lf", kq1);
83 printf("\nkq2 = %f", kq2);
84 printf("\nkq3 = %d", kq3);
85 }
86
```


kq1 = 2.828427

kq2 = 2.828427

kq3 = 2

C:\Users\nvdangkhoa\Desktop\LapTrinhBanBan\Debug\LapTrinhBanBan.exe (process 20580) exited with code 0.

Press any key to close this window . . .

```
71
72 //printf("\n\nKy tu: c = %c", c);
73 //printf("\nSo nguyen: c = %d", c);
74 //printf("\nKy tu in thuong: c = %c", c + 32);
75
76 int n = 8;
77
78 double kq1 = sqrt((double)n);
79 float kq2 = (float) sqrt((double)n);
80 int kq3 = (int) sqrt((double)n);
81
82 printf("\nkq1 = %lf", kq1);
83 printf("\nkq2 = %f", kq2);
84 printf("\nkq3 = %d", kq3);
85
86 double kq4 = pow((double)2, (double)3);
87 int kq5 = (int) pow((double)2, (double)3);
88
89 printf("\nkq4 = %lf", kq4);
90 printf("\nkq5 = %d", kq5);
91 }
92
```

a. Kiểu số nguyên:

Kiểu số nguyên thường được thực hiện với các phép toán: +, -, *, /, %, >, >=, <, <=, ==, ...

■ Các kiểu số nguyên có dấu

khi ép kiểu phải chú ý:

- ép từ số ít lên số nhiều thì ok

- ép từ số nhiều lên số ít sẽ dễ mất dữ liệu

Ví dụ:

- char ít số hơn short, int, long => ép lên thì ổn

- long nhiều số hơn các kiểu còn lại, ép xuống thì dễ mất dữ liệu



char	1	-128 ... +127
short	2	-32768 ... +32767
int	2	-32768 ... +32767
long	4	-2147483648 ... +2147483647