



eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



C-Basics quiz solution

KEROLES SHENOUDA





#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda
<https://www.facebook.com/groups/er...>

Learn-in-depth progress Sequence



Start Quiz

Start quiz

Start Study



www.Learn-in-depth.com



Put your understanding level

Solve the assignments

Forum > Embedded topics

Embedded Topics

here you can share any embedded topic you have

Title	Recent Activity
C Basics session 1 assignment	a few seconds ago

Sort by: Recent Activity Create New Post

<https://www.facebook.com/groups/embedded.system.KS/>





C Basics Tricks part 1

1
Operators priorities & Implicit Type Conversion

```
float g = 2 / 3.0 + 5 ;
g=5.6
float g = 2 / 3 + 5 ;
g=5
```

2 Signed Data type Range (+/-)
And Two's complement

assume that a is int with 2 bytes*

The expression, a = 30 * 1000 + 2768 ; evaluates to

- (1) 32768
- (2) -32768
- (3) 113040
- (4) 0

3

The sign of the result of a remainder operation, according to C99, is the same as the dividend's one.

Let's see some examples (dividend / divisor):

The expression x = 4 + 2 % -8 evaluates to

- (1) -6
- (2) 6
- (3) 4
- (4) None of the above

4 Ascii with Octa number

```
#include <stdio.h>
int main()
{
    char a = '\012';
    printf("%d", a);
    return 0;
}
```

10

5

a==b==c (Multiple Comparison) evaluates in C programming
(==) operates from left to right
Expression a==b==c is actually (a==b) ==c,

if(a==b==c)

6

Optimization will ignore the operation

```
#include <stdio.h>
int main()
{
    int i = 5, j = 10, k = 15;
    printf("%d ", sizeof(k /= i + j));
    printf("%d", k);
    return 0;
}
```

Ans. 4 15

9

Prefix and postfix

```
#include <stdio.h>
int main()
{
    int x = 10;
    int y = (x++, x++, x++);
    printf("%d %d\n", x, y);
    return 0;
}
```

13 12

7

Int i= 1,2,3 → i= 3

8

Bit shifting is *not* multiplication. It can be used in certain circumstances to have the same effect as a multiplication by a power of two
addition(+) operator has higher precedence than shift(<<) operator

```
{
    printf("%d", 1 << 2 + 3 << 4);
    return 0;
    1<<5<<4 == 1<<9 == 1*2^9 =512
}
```

10

Set Bit

X |= 1<<n ;

Reset specific Bit

X &= ~ (1<<n);

Toggle Specific Bit

X ^= 1<<n ;

11

Implicit/
explicit
tricks

12

In case of GCC Compiler
Output will be 12,10. Output
may very from compiler to compiler because order of
evaluation inside printf

```
#include <stdio.h>
int main()
{
    int i = 10;
    printf("%d, %d\n", ++i, i++);
    return 0;
}
```

The reason for **undefined behavior** in PROGRAM 1 is,
the operator '+' doesn't have
standard defined order of
evaluation for its operands

```
#include <stdio.h>
int main(void)
{
    int a = 1;
    int b = 0;
    b = a++ + a++;
    printf("%d %d", a, b);
}
```

compiler Dependent





#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

4

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

q1

Which of the following are invalid variable names ? *

- BASICSALARY
- #MEAN
- group
- 422
- hELLO
- queue
- FLOAT
- Plot # 3
- _basic

Which of the following are invalid variable names ?

- BASICSALARY
- #MEAN
- group
- 422
- hELLO
- queue
- FLOAT
- Plot # 3
- _basic
- Add option or ADD "OTHER"

Checkboxes

X X X X X X X X X X



q2

► Evaluate the following expressions

- $g = \text{big} / 2 + \text{big} * 4 / \text{big} - \text{big} + \text{abc} / 3 ; (\text{abc} = 2.5, \text{big} = 2, \text{assume } g \text{ to be a float})$
- $\text{on} = \text{ink} * \text{act} / 2 + 3 / 2 * \text{act} + 2 + \text{tig} ; (\text{ink} = 4, \text{act} = 1, \text{tig} = 3.2, \text{assume } \text{on} \text{ to be an int})$
- $s = \text{qui} * \text{add} / 4 - 6 / 2 + 2 / 3 * 6 / \text{god} ; (\text{qui} = 4, \text{add} = 2, \text{god} = 2, \text{assume } s \text{ to be an int})$

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

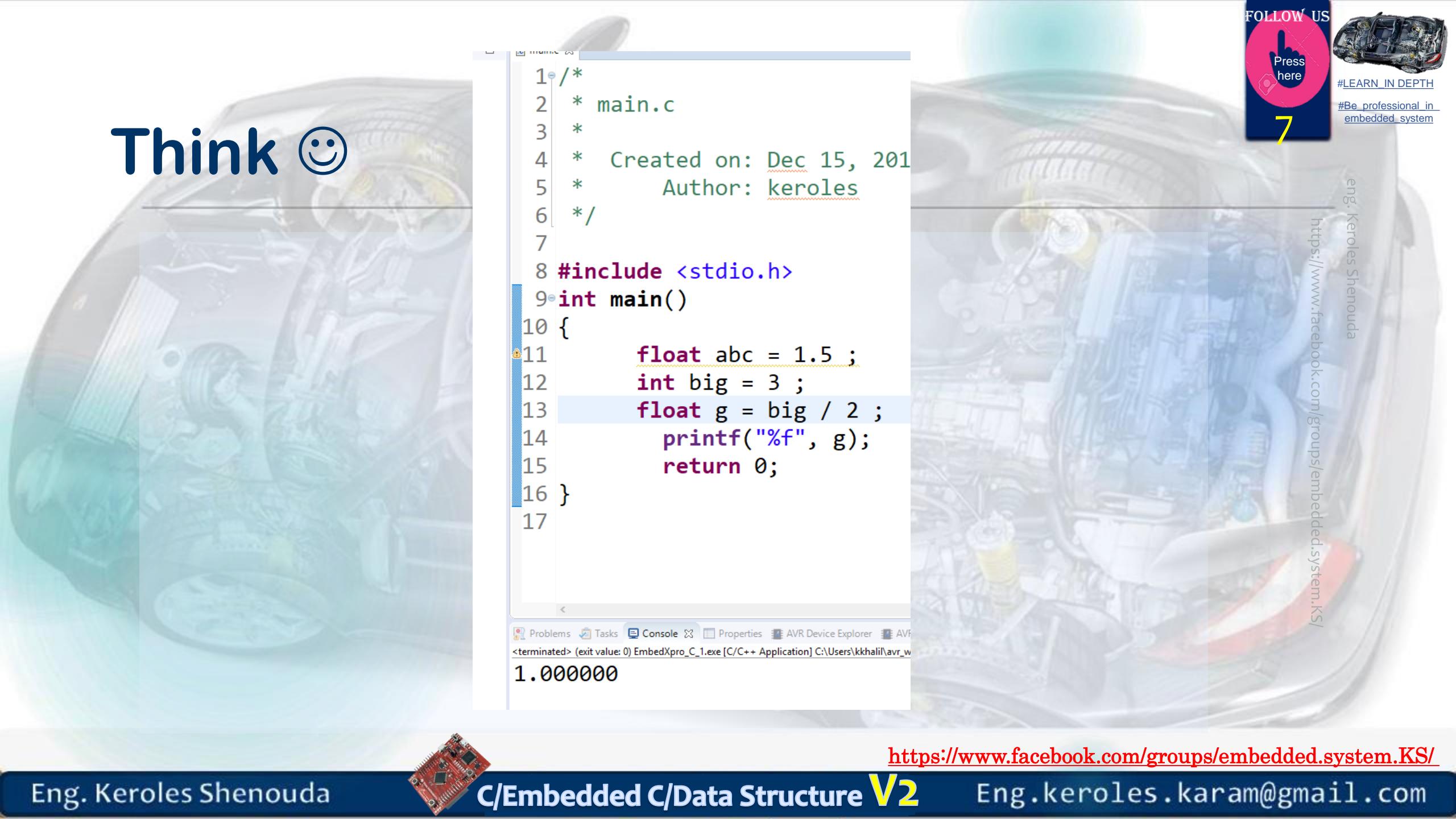
Think ☺

```
1 /*  
2 * main.c  
3 *  
4 *   Created on: Dec 15, 2018  
5 *     Author: keroles  
6 */  
7  
8 #include <stdio.h>  
9 int main()  
10 {  
11     float abc = 1.5 ;  
12     int big = 3 ;  
13     float g = big / 2 ;  
14     printf("%f", g);  
15     return 0;  
16 }  
17
```

<https://www.facebook.com/groups/embedded.system.KS/>



Think ☺



```
1 /*  
2  * main.c  
3  *  
4  * Created on: Dec 15, 201  
5  * Author: keroles  
6 */  
7  
8 #include <stdio.h>  
9 int main()  
10 {  
11     float abc = 1.5 ;  
12     int big = 3 ;  
13     float g = big / 2 ;  
14     printf("%f", g);  
15     return 0;  
16 }  
17
```

Problems Tasks Console Properties AVR Device Explorer AVR
<terminated> (exit value: 0) EmbedXpro_C_1.exe [C/C++ Application] C:\Users\kkhalil\avr_w
1.000000



eng.Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





Think

```
1/*  
2 * main.c  
3 *  
4 * Created on: Dec 15, 2018  
5 * Author: keroles  
6 */  
7  
8#include <stdio.h>  
9int main()  
10{  
11    float abc = 1.5 ;  
12    int big = 3 ;  
13    float g = big / 2.0 ;  
14    printf("%f", g);  
15    return 0;  
16}  
17
```

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

```
main.c X
1  /*
2   * main.c
3   *
4   * Created on: Dec 15, 2018
5   * Author: keroles
6   */
7
8 #include <stdio.h>
9 int main()
10 {
11     float abc = 1.5 ;
12     int big = 3 ;
13     float g = big / 2.0 ;
14     printf("%f", g);
15     return 0;
16 }
17
```

Problems Tasks Console Properties AVR Device Explorer AVR Supported I
<terminated> (exit value: 0) EmbedXpro_C_1.exe [C/C++ Application] C:\Users\kkhalil\avr_ws\EmbedXpr
1.500000

<https://www.facebook.com/groups/embedded.system.KS/>

q2

Two tricks

Int/int = int

Float/int = float

Operation priorities

► Evaluate the following expressions

- $g = \text{big} / 2 + \text{big} * 4 / \text{big} - \text{big} + \text{abc} / 3$; ($\text{abc} = 1.5$, $\text{big} = 3$, assume g to be a float)
- $on = \text{ink} * \text{act} / 2 + 3 / 2 * \text{act} + 2 + \text{tig}$; ($\text{ink} = 3$, $\text{act} = 2$, $\text{tig} = 3.2$, assume on to be an int)
- $s = \text{qui} * \text{add} / 4 - 6 / 2 + 2 / 3 * 6 / \text{god}$; ($\text{qui} = 2$, $\text{add} = 4$, $\text{god} = 3$, assume s to be an int)

```

g = 3/2 + 3*4/3 - 3 + 1.5/
3
g = 1 + 3 * 4/3 - 3 + 1.5/3
g = 1 + 12/3 - 3 + 1.5/3
g = 1 + 4 - 3 + 1.5/3
g = 1 + 4 - 3 + 0.5
g = 5 - 3 + 0.5
g = 2 + 0.5
g = 2.5
    
```

operation: /

operation: *

operation: /

operation: /

operation: +

operation: -

operation: +

```

on = 3 * 2 / 2 + 3 / 2 * 2 + 2
+ 3.2
on = 6 / 2 + 3 / 2 * 2 + 2 +
3.2
on = 3 + 3 / 2 * 2 + 2 + 3.2
on = 3 + 1 * 2 + 2 + 3.2
on = 3 + 2 + 2 + 3.2
on = 5 + 2 + 3.2
on = 7 + 3.2
on = 10
    
```

operation: *

operation: /

operation: /

operation: *

operation: +

operation: +

operation: +

```

s = 2 * 4 / 4 - 6 / 2 + 2 / 3 *
6 / 3
s = 8 / 4 - 6 / 2 + 2 / 3 * 6 /
3
s = 2 - 6 / 2 + 2 / 3 * 6 / 3
s = 2 - 3 + 2 / 3 * 6 / 3
s = 2 - 3 + 0 * 6 / 3
s = 2 - 3 + 0 / 3
s = 2 - 3 + 0
s = -1 + 0
s = -1
    
```

operation: *

operation: /

operation: /

operation: *

operation: /

operation: -

operation: +



q3

Two tricks

Int/int = int

Float/int = float

Operation priorities

```
int i = 2, j = 3, k, l;  
float a, b;  
k = i / j * j;  
l = j / i * i;  
a = i / j * j;  
b = j / i * i;  
printf ("%d %d %f %f", k, l, a, b);
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



q3

Two tricks

Int/int = int

Float/int = float

Operation priorities

```
int i = 2, j = 3, k, l;  
float a, b;  
k = i / j * j;  
l = j / i * i;  
a = i / j * j;  
b = j / i * i;  
printf ("%d %d %f %f", k, l, a, b);
```

0 2 0.0 2.0

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



Q4 no tricks

programs are converted into machine language with the help of

- (1) An interpreter
- (2) A compiler
- (3) An operating system
- (4) None of the above

Compiler

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda





Q5/6/7/8 No tricks

The real constant in C can be expressed in which of the following forms

- (1) Fractional form only
- (2) Exponential form only
- (3) ASCII form only
- (4) Both fractional and exponential forms

In $b = 6.6 / a + (2 * a + (.3 * c) / a * d) / (2 / n)$; which operation will be performed first?

- (1) $6.6 / a$
- (2) $2 * a$
- (3) $.3 * c$
- (4) $2 / n$

) Which of the following statement is wrong

- (1) `mes = 123.56 ;`
- (2) `con = 'T' * 'A' ;`
- (3) `this = 'T' * 20 ;`
- (4) `3 + a = b ;`

Which of the following shows the correct hierarchy of arithmetic operations in C

- (1) `(), **, * or /, + or -`
- (2) `(), **, *, /, +, -`
- (3) `(), **, /, *, +, -`
- (4) `(), / or *, - or +`



Q9 no tricks

If a is an integer variable, $a = 5 / 2$; will return a value

- (1) 2.5
- (2) 3
- (3) 2
- (4) 0

The expression, $a = 7 / 22 * (3.14 + 2) * 3 / 5$; evaluates to

- (1) 8.28
- (2) 6.28
- (3) 3.14
- (4) 0



Q10

Two tricks

Int/int = int

Float/int = float

Operation priorities

The expression, $a = 7 / 22 * (3.14 + 2) * 3 / 5$; evaluates to

- (1) 8.28
- (2) 6.28
- (3) 3.14
- (4) 0

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Q10

Two tricks

Int/int = int

Float/int = float

Operation priorities

The expression, $a = 7 / 22 * (3.14 + 2) * 3 / 5$; evaluates to

- (1) 8.28
- (2) 6.28
- (3) 3.14
- (4) 0

$$\begin{aligned} & 7/22*5.14*3/5 \\ & 0*5.14*3/5 \end{aligned}$$

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



Q11

One trick

Signed int Range for 2 bytes

assume that a is int with 2 bytes *

The expression, $a = 30 * 1000 + 2768$; evaluates to

- (1) 32768
- (2) -32768
- (3) 113040
- (4) 0

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Q11

One trick

Signed int Range for **2 bytes**

assume that a is int with 2 bytes *

The expression, $a = 30 * 1000 + 2768$; evaluates to

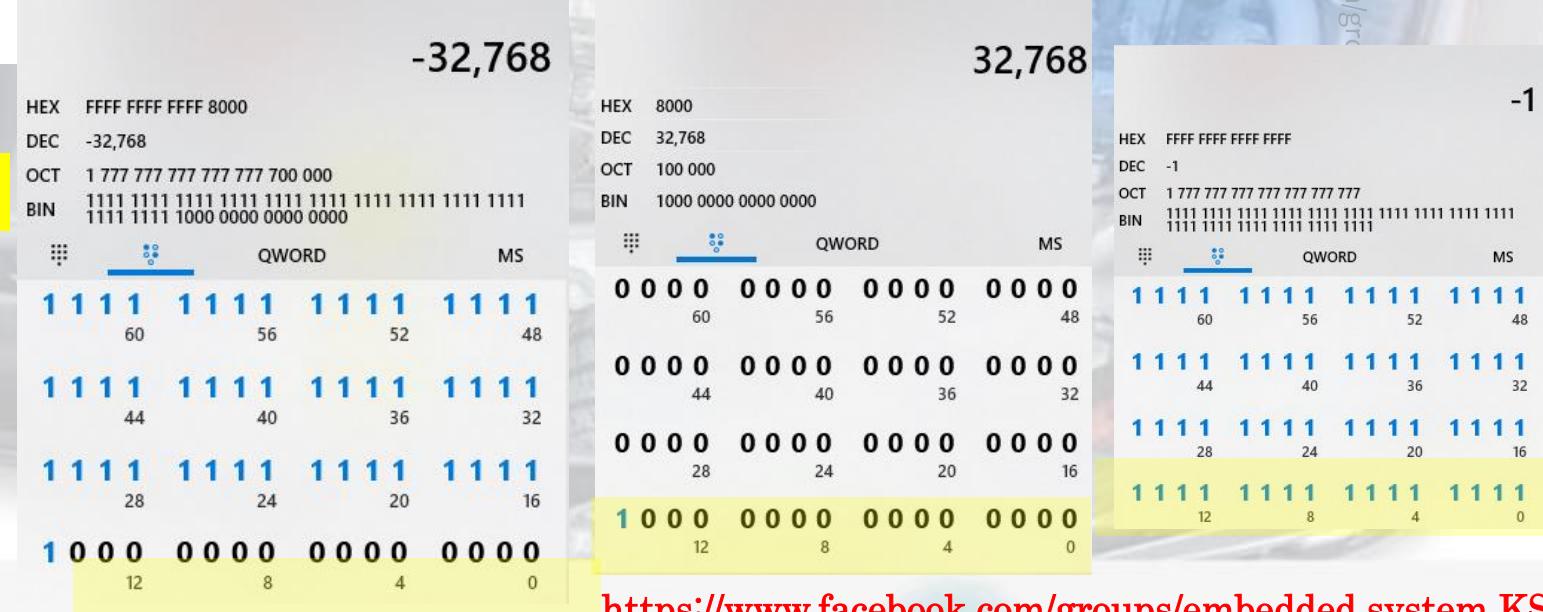
- (1) 32768
- (2) -32768
- (3) 113040
- (4) 0

First calculate the expression = 32768

1	0	0	0	0	0	0	0
12	8	4	0				

If size of a signed data type is n bytes, it ranges from -2^{8n-1} to $2^{8n-1}-1$

So, a short(usually 2 bytes) ranges from **-32768** to **32767** and an unsigned short ranges from 0 to 65535



<https://www.facebook.com/groups/embedded.system.KS/>



Q12

The expression $x = 4 + 2 \% -8$ evaluates to

- (1) -6
- (2) 6
- (3) 4
- (4) None of the above

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Think ☺

```

8 #include <stdio.h>
9 int main()
10 {
11     float abc = 1.5 ;
12     int big = 3 ;
13     int g1 = 5 % (-3) ;
14     int g2 = (-5) % (3);
15     int g3 = (-5) % (-3);
16     printf("%d %d %d \n", g1,g2,g3);
17     g1 = 5 / (-3) ;
18     g2 = (-5) / (3);
19     g3 = (-5) / (-3);
20     printf("%d %d %d", g1,g2,g3);
21     return 0;
22 }
```

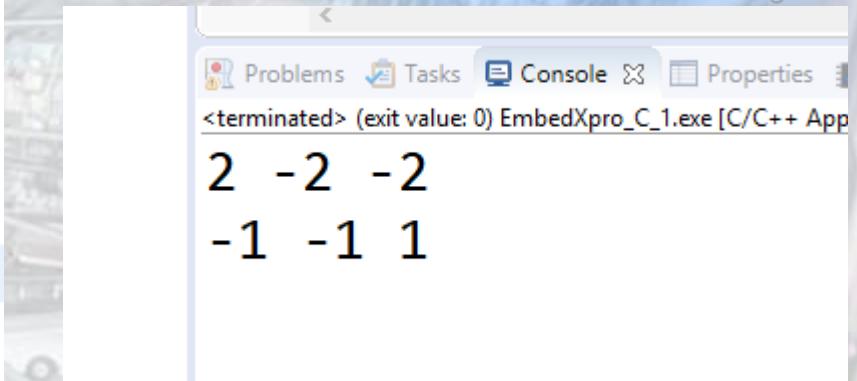
<https://www.facebook.com/groups/embedded.system.KS/>



Think ☺

```

8 #include <stdio.h>
9 int main()
10 {
11     float abc = 1.5 ;
12     int big = 3 ;
13     int g1 = 5 % (-3) ;
14     int g2 = (-5) % (3);
15     int g3 = (-5) % (-3);
16     printf("%d %d %d \n", g1,g2,g3);
17     g1 = 5 / (-3) ;
18     g2 = (-5) / (3);
19     g3 = (-5) / (-3);
20     printf("%d %d %d", g1,g2,g3);
21     return 0;
22 }
```



Problems Tasks Console Properties

<terminated> (exit value: 0) EmbedXpro_C_1.exe [C/C++ App]

2 -2 -2
-1 -1 1

eng.Keroles Shenouda
<https://www.facebook.com>



Q12

One trick

The sign of the result of a remainder operation, according to C99, is the same as the dividend's one.

Let's see some examples (dividend / divisor):

The expression $x = 4 + 2 \% -8$ evaluates to

- (1) -6
- (2) 6
- (3) 4
- (4) None of the above

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



q13

Assume that the size of char is 1 byte and negatives are stored in 2's complement form

```
#include<stdio.h>
int main()
{
    char c = 125;
    c = c+10;
    printf("%d", c);
    return 0;
}
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



q13

One trick

Signed Data type Range (+/-)

Assume that the size of char is 1 byte and negatives are stored in 2's complement form

```
#include<stdio.h>
int main()
{
    char c = 125;
    c = c+10;
    printf("%d", c);
    return 0;
}
```

Char	Integer	1	1	-128 to 127
------	---------	---	---	-------------

HEX	87	DEC	135	OCT	207	BIN	1000 0111	QWORD	MS
0	000	000	000	000	000	000	000	000	000
60		56		52		48			
0	000	000	000	000	000	000	000	000	000
44		40		36		32			
0	000	000	000	000	000	000	000	000	000
28		24		20		16			
0	000	000	000	000	000	000	1000 0111	0	0
12		8		4		0			

-X

Two's complement

0111 1001 (+121)

-X will be -121

<https://www.facebook.com/groups/embedded.system.KS/>



q14

Predict the output of following program. Assume that the numbers are stored * in 2's complement form.

```
#include<stdio.h>
int main()
{
    unsigned int x = -1;
    int y = ~0;
    if (x == y)
        printf("same");
    else
        printf("not same");
    return 0;
}
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



q14

Predict the output of following program. Assume that the numbers are stored * in 2's complement form.

```
#include<stdio.h>
int main()
{
    unsigned int x = -1;
    int y = ~0;
    if (x == y)
        printf("same");
    else
        printf("not same");
    return 0;
}
```

X=0xFFFF FFFF

y=0xFFFF FFFF

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Q15 one trick

Int/int = int

Float/int = float

Operation priorities

Predict the output *

```
#include <stdio.h>

int main()
{
    float c = 5.0;
    printf ("Temperature in Fahrenheit is %.2f", (9/5)*c + 32);
    return 0;
}
```

- (A) Temperature in Fahrenheit is 41.00
- (B) Temperature in Fahrenheit is 37.00
- (C) Temperature in Fahrenheit is 0.00
- (D) Compiler Error

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Q15 one trick

Int/int = int

Float/int = float

Operation priorities

Predict the output *

```
#include <stdio.h>

int main()
{
    float c = 5.0;
    printf ("Temperature in Fahrenheit is %.2f", (9/5)*c + 32);
    return 0;
}
```

- (A) Temperature in Fahrenheit is 41.00
- (B) Temperature in Fahrenheit is 37.00
- (C) Temperature in Fahrenheit is 0.00
- (D) Compiler Error

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





30

q16

```
#include <stdio.h>
int main()
{
    char a = '\012';
    printf("%d", a);
    return 0;
}
```

- (A) Compiler Error
- (B) 12
- (C) 10
- (D) Empty

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

q16

One trick

Asci with Octa number

```
#include <stdio.h>
int main()
{
    char a = '\012';
    printf("%d", a);
    return 0;
}
```

- (A) Compiler Error
- (B) 12
- (C) 10
- (D) Empty

Programmer

HEX	A
DEC	10
OCT	12
BIN	1010

		QWORD				MS	
		Lsh	Rsh	Or	Xor	Not	And
↑	Mod	CE	C	⊗	÷		
A	B	7	8	9	X		
C	D	4	5	6	-		
E	F	1	2	3	+		
()	±	0	.	=		

<https://www.facebook.com/groups/embedded.system.KS/>

eng.Keroles Shenouda



q17

Predict the output of the below program: *

```
#include <stdio.h>
int main()
{
    printf("%d", 1 << 2 + 3 << 4);
    return 0;
}
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





Think 😊

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

```
1/*  
2 * main.c  
3 *  
4 * Created on: Dec 15, 2018  
5 * Author: keroles  
6 */  
7  
8 #include <stdio.h>  
9 int main()  
10 {  
11 |  
12     printf("%d \n", 1 + 2 * 8);  
13     printf("%d \n", 1 + 2 << 3);  
14  
15     return 0;  
16 }  
17  
18 }
```

<https://www.facebook.com/groups/embedded.system.KS/>



Think 😊

```
1/*  
2 * main.c  
3 *  
4 * Created on: Dec 15, 2018  
5 * Author: keroles  
6 */  
7  
8 #include <stdio.h>  
9 int main()  
10 {  
11 |  
12     printf("%d \n", 1 + 2 * 8);  
13     printf("%d \n", 1 + 2 << 3);  
14  
15  
16         return 0;  
17 }  
18
```

Problems Tasks Console Properties AVR Device Explorer AVR Supported MCUs
<terminated> (exit value: 0) EmbedXpro_C_1.exe [C/C++ Application] C:\Users\kkhalil\avr_ws\EmbedXpro_C_1\Debug\EmbedXpro_C_1.exe (4/12/1)
17
24



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

34



q17

One trick

Bit shifting is *not* multiplication. It can be used in certain circumstances to have the same effect as a multiplication by a power of two

Predict the output of the below program: *

```
#include <stdio.h>
int main()
{
    printf("%d", 1 << 2 + 3 << 4);
    return 0;
}
```

$$1 \ll 5 \ll 4 == 1 \ll 9 == 1 * 2^9 = 512$$

Explanation: The main logic behind the program is the precedence and associativity of the operators. The addition(+) operator has higher precedence than shift(<<) operator. So, the expression boils down to $1 \ll (2 + 3) \ll 4$ which in turn reduces to $(1 \ll 5) \ll 4$ as the shift operator has left-to-right associativity



q18

One trick

Int i= 1,2,3 == i= 3

```
#include <stdio.h>

int main()
{
    int i = (1, 2, 3);

    printf("%d", i);

    return 0;
}
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





37

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

q19

One trick

Optimization tricks

```
#include <stdio.h>
int main()
{
    int i = 5, j = 10, k = 15;
    printf("%d ", sizeof(k /= i + j));
    printf("%d", k);
    return 0;
}
```

4 15





38

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

q20

```
#include <stdio.h>
int main()
{
    int a = 10, b = 20, c = 30;
    if (c > b > a)
        printf("TRUE");
    else
        printf("FALSE");
    return 0;
}
```



Eng. Keroles Shenouda

C/Embedded C/Data Structure V2

<https://www.facebook.com/groups/embedded.system.KS/>

Eng.keroles.karam@gmail.com



39

Think ☺

```
1 /*  
2  * main.c  
3  *  
4  * Created on: Dec 15, 2018  
5  * Author: keroles  
6 */  
7  
8 #include <stdio.h>  
9 int main(){  
10    int a,b,c;  
11    a=b=c=100;  
12  
13    if(a==b==c)  
14        printf("True...\n");  
15    else  
16        printf("False...\n");  
17  
18    return 0;  
19 }  
20
```

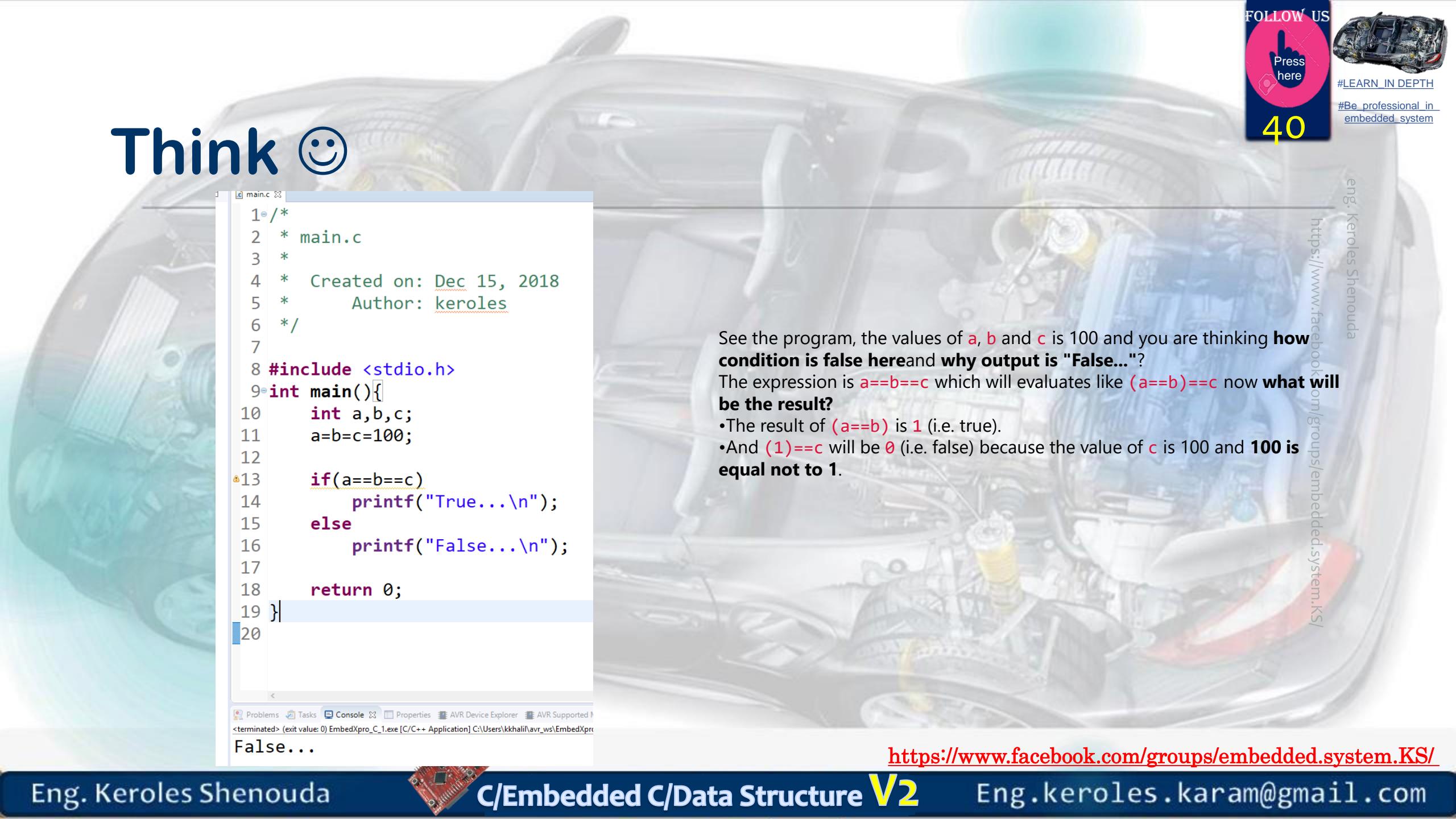
<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



Think 😊



```
main.c
1  /*
2  * main.c
3  *
4  * Created on: Dec 15, 2018
5  * Author: keroles
6  */
7
8 #include <stdio.h>
9 int main(){
10     int a,b,c;
11     a=b=c=100;
12
13     if(a==b==c)
14         printf("True...\n");
15     else
16         printf("False...\n");
17
18     return 0;
19 }
20
```

False...

Problems Tasks Console Properties AVR Device Explorer AVR Supported <terminated> (exit value: 0) EmbedXpro_C_1.exe [C/C++ Application] C:\Users\kkhalil\avr_ws\EmbedXpro

See the program, the values of **a**, **b** and **c** is 100 and you are thinking **how condition is false here and why output is "False..."?**

The expression is **$a==b==c$** which will evaluates like **$(a==b)==c$** now **what will be the result?**

- The result of **$(a==b)$** is **1** (i.e. true).
- And **$(1)==c$** will be **0** (i.e. false) because the value of **c** is 100 and **100 is equal not to 1**.

<https://www.facebook.com/groups/embedded.system.KS/>



q20

One trick

a==b==c (Multiple Comparison) evaluates in C programming
(==) operates from left to right
Expression a==b==c is actually (a==b)==c,

```
#include <stdio.h>
int main()
{
    int a = 10, b = 20, c = 30;
    if (c > b > a)
        printf("TRUE");
    else
        printf("FALSE");
    return 0;
}
```

Output:
FALSE

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



q21

```
#include <stdio.h>
int main()
{
    int a = 0;
    int b;
    a = (a == (a == 1));
    printf("%d", a);
    return 0;
}
```

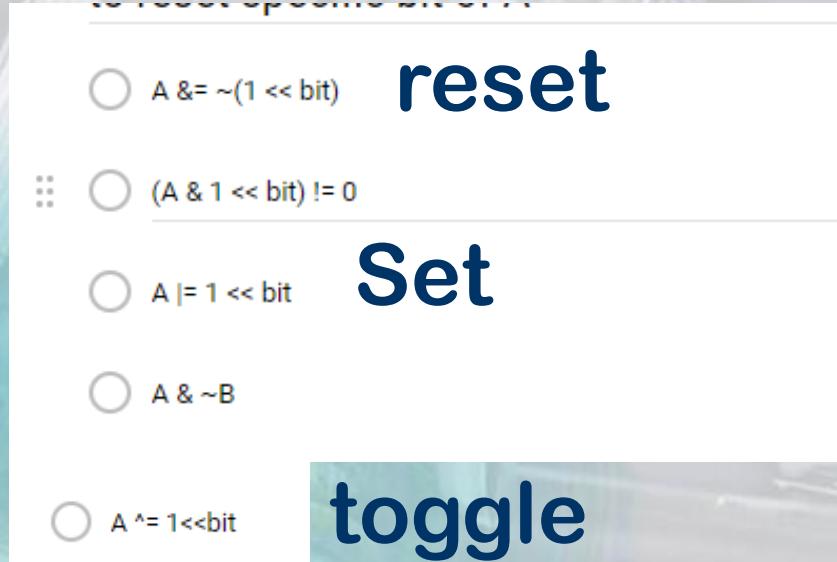
= 1

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



Q22/23/24



eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





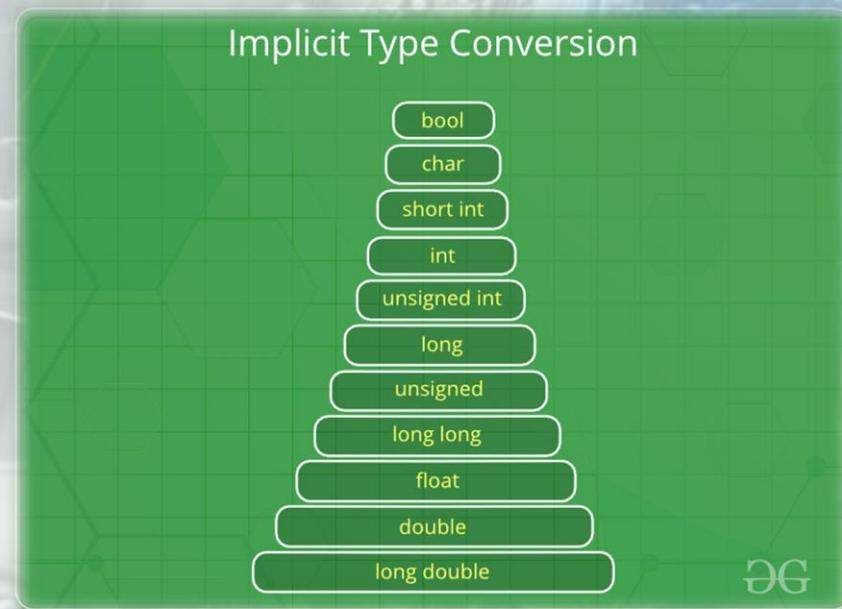
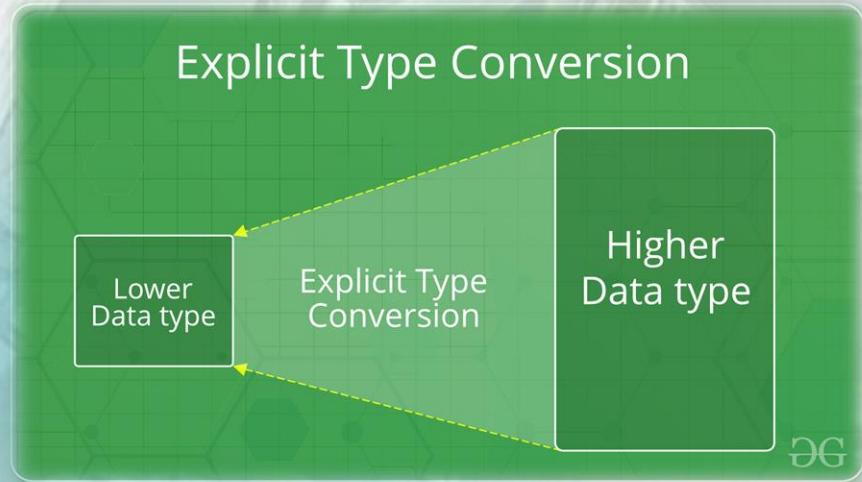
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Implicit/explicit tricks



Type Conversion in C



eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>





Press here

#LEARN IN DEPTH

#Be_professional_in_embedded_system

46

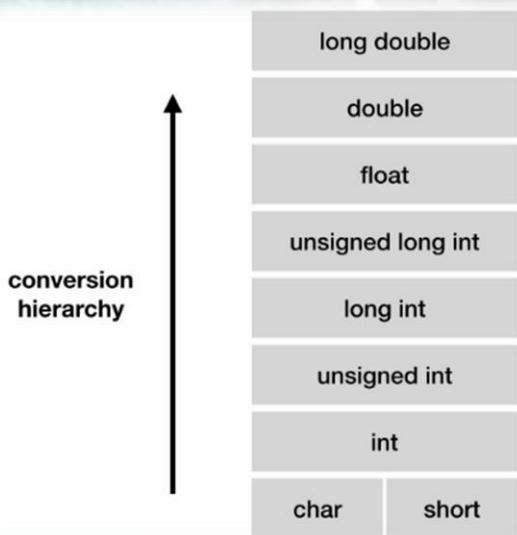
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Implicit Type Conversion

Also known as 'automatic type conversion'.

- Done by the compiler on its own, without any external trigger from the user.
- Generally takes place when in an expression more than one data type is present. In such condition type conversion (type promotion) takes place to avoid lose of data.
- All the data types of the variables are upgraded to the data type of the variable with largest data type.
- It is possible for implicit conversions to lose information, signs can be lost (when signed is implicitly converted to unsigned), and overflow can occur (when long long is implicitly converted to float).



```

// An example of implicit conversion
#include<stdio.h>
int main()
{
    int x = 10;      // integer x
    char y = 'a';   // character c

    // y implicitly converted to int. ASCII
    // value of 'a' is 97
    x = x + y;

    // x is implicitly converted to float
    float z = x + 1.0;

    printf("x = %d, z = %f", x, z);
    return 0;
}
  
```

x = 107, z = 108.00000

<https://www.facebook.com/groups/embedded.system.KS/>





Press
here

#LEARN IN DEPTH
#Be_professional_in
embedded_system

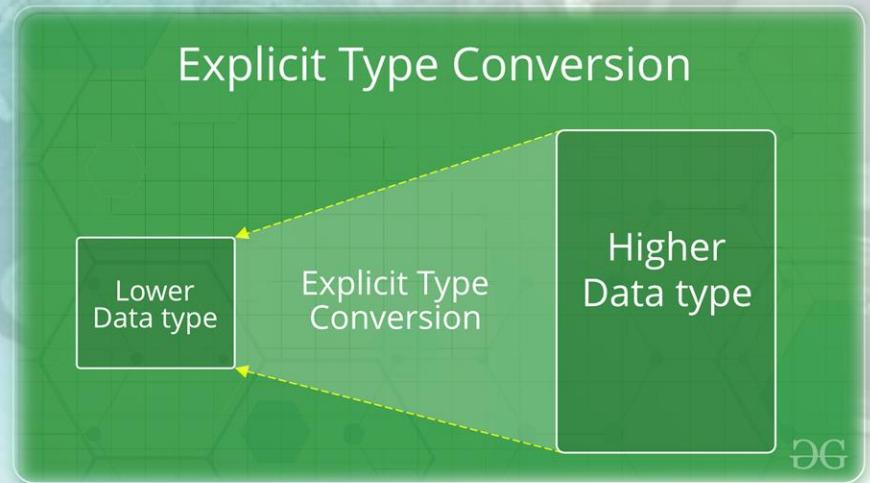
47

Explicit Type Conversion

This process is also called type casting and it is user defined. Here the user can type cast the result to make it of a particular data type.

The syntax in C:

(type) expression



```
// C program to demonstrate explicit type casting
#include<stdio.h>

int main()
{
    double x = 1.2;

    // Explicit conversion from double to int
    int sum = (int)x + 1;

    printf("sum = %d", sum);

    return 0;
}
```

sum = 2

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



Implicit/explicit tricks

.....Conversion is done programatically. *

- Implicit
- Explicit
- other

Conversion of larger number to smaller number is conversion.float
k=123.456int i= (int) k

- Implicit
- Explicit
- other

type casting is to *

- Implicit
- Explicit
- other

.....Conversion is done automatically.

- Implicit
- Explicit
- other

Conversion of smaller number to larger number isconversion.Conversion
of integer type data to float.float i=0;int j=10;i=j;

- Implicit
- Explicit
- other



q30

The reason for undefined behavior in PROGRAM 1 is, the operator '+' doesn't have standard defined order of evaluation for its operands

```
#include<stdio.h>
int main(void)
{
    int a = 1;
    int b = 0;
    b = a++ + a++;
    printf("%d %d",a,b);
    return 0;
}
```

- 3 6
- compiler Dependent
- 3 4
- 3 3

eng. Keroles Shenouda

<https://www.ks.com.eg>



Explanation: See <https://www.geeksforgeeks.org/sequence-points-in-c-set-1/> for explanation.

system.KS/





50

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

q31

```
#include <stdio.h>
int main()
{
    int x = 10;
    int y = (x++, x++, x++);
    printf("%d %d\n", x, y);
    return 0;
}
```

- 13 12
- compiler Dependent
- 13 13
- 10 10





q31

One trick

Prefix and postfix

```
#include <stdio.h>
int main()
{
    int x = 10;
    int y = (x++, x++, x++);
    printf("%d %d\n", x, y);
    return 0;
}
```

13 12

compiler Dependent

13 13

10 10

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



q32

One trick

Prefix and postfix

```
#include <stdio.h>
int main()
{
    int i = 3;
    printf("%d", (++i)++);
    return 0;
}
```

- 3
- 4
- 5
- compile error

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



q32

One trick

Prefix and postfix

```
#include <stdio.h>
int main()
{
    int i = 3;
    printf("%d", (++i)++);
    return 0;
}
```

- 3
- 4
- 5
- compile error

Explanation: In C, prefix and postfix operators

need **I-value** to perform operation and return

r-value. The expression $(++i)++$ when executed increments the value of variable i (i is a I-value) and returns r-value. The compiler generates the error(I-value required) when it tries to post-incremeny the value of a r-value.

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



q33

In case of GCC Compiler Output will be 12,10. Output may vary from compiler to compiler because order of evaluation inside printf

Predict the output of the following code ?

```
#include<stdio.h>
int main()
{
    int i = 10;
    printf("%d, %d\n", ++i, i++);
    return 0;
}
```

- 12 10
- 12 12
- 12 11
- Output may Vary from Compiler to Compiler

<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



q34

```
int X,i=4,j=7; X=j || i++ && 1
```

X=1 , i =4

X=7 , i =4

X=7 , i =5

compile error

$X = (j \parallel (i++ \&& 1));$

Note that precedence does not equate to order of execution in general. In this case, we have the following evaluation logic:

To evaluate = we need to evaluate its right-hand operand

To evaluate (j || stuff...) we first evaluate j

j is non-zero, so the result of (j || stuff...) is 1, and we do not evaluate stuff due to the short-circuit behaviour of ||

Now we have determined that the right-hand operand of == has evaluated to 1, so assign 1 to X.

Final result: X == 1, and i and j unchanged.



q35

```
void main()
{
    int i=0, j=1, k=2, m;
    m = i++ || j++ || k++;
    printf("%d %d %d %d", m, i, j, k);
}
```

- 1123
- 1122
- 0122
- 0123
- None of these

<https://www.facebook.com/groups/embedded.system.KS/>



q35

```
void main()
{
    int i=0, j=1, k=2, m;
    m = i++ || j++ || k++;
    printf("%d %d %d %d", m, i, j, k);
}
```

- 1 1 2 3
- 1 1 2 2
- 0 1 2 2
- 0 1 2 3
- None of these

Answer: Option B

Solution:

In an expression involving || operator, evaluation takes place from left to right and will be stopped if one of its components evaluates to true(a non zero value).

So in the given expression $m = i++ || j++ || k++$.

It will be stop at j and assign the current value of j in m . therefore $m = 1$, $i = 1$, $j = 2$ and $k = 2$ (since $k++$ will not encounter. so its value remain 2)



q36

. What will be the output of the following code fragment?

```
void main ()
{
    printf("%x", -1<<4);
}
```

- fff0
- fff1
- fff2
- fff3
- fff4

Answer: Option A

Solution:

-1 will be represented in binary form as:

1111 1111 1111 1111

Now -1<<4 means 1 is Shifted towards left by 4 positions, hence it becomes:

1111 1111 1111 0000 in hexadecimal form - fff0.



Thanks and Good Luck



ENG. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.facebook.com/groups/embedded.system.KS/>

