C Basics Part 1 QUESTIONS

Total points 60/60

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

Email *

abdallah.shabaan.ghazy@gmail.com

14 of 14 points

✓ assume that a is int with 2 bytes * 1/1

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

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

- ○ 1
- ◉ 2 ✓
- ○ 3
- ○ 4

✓ *

1/1

) Which of the following statement is wrong

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

- 1
- 2
- 3
- (●) 4 ✓

✓ *

1/1

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

- 1
- 2
- 3
- (●) 4 ✓

✓ * 1/1

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

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

- ○ 1
- ◉ 2 ✓
- ○ 3
- ○ 4

✓ What would be the output of the following programs: * 1/1

```
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 2.0 0.0
- ○ 0 2 2.0 2.0
- ◉ 0 2 0.0 2.0 ✓
- ○ 2 0 2.0 0.0

✓ *

1/1

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

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

- 1
- 2
- 3
- (●) 4 ✓

✓ *

1/1

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

- 1
- 2
- (●) 3 ✓
- 4

Evaluate the following expressions *

	-1	10	2.5	Score	
g = big / 2 + big * 4 / big - big + abc / 3 ; (abc =1.5, big = 3, assume g to be a float)	○	○	◉	1/1	✓
on = ink * act / 2 + 3 / 2 * act + 2 + tig ; (ink = 3, act =2, tig = 3.2, assume on to be an int)	○	◉	○	1/1	✓
s = qui * add / 4 - 6 / 2 + 2 / 3 * 6 / god ; (qui = 2, add = 4, god =3, assume s to be an int)	◉	○	○	1/1	✓

✓ *

1/1

C 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

- ○ 1
- ◉ 2 ✓
- ○ 3
- ○ 4

✓ *

1/1

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

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

- ○ 1
- ○ 2
- ○ 3
- ◉ 4 ✓

✓ Which of the following are invalid variable names ? * 1/1

- [ ] BASICSALARY
- [x] #MEAN ✓
- [ ] group
- [x] 422 ✓
- [ ] hELLO
- [ ] queue
- [ ] FLOAT
- [x] Plot # 3 ✓
- [ ] _basic

✓ * 1/1

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

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

- ○ 1
- ○ 2
- ◉ 3 ✓
- ○ 4

C Basics tricks 9 of 9 points

✓ Predict the output of following program. Assume that the numbers are stored in 2's complement form. *1/1

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

- (●) same ✓
- ( ) not same

✓ * 1/1

```
#include <stdio.h>
int main()
{
    char a = '\012';

    printf("%d", a);

    return 0;
}
```

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

✓ * 1/1

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

- 41
- (●) 4 15 ✓
- 2 1
- compile error

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

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

- 135
- +INF
- (●) -121 ✓
- -8

✓ 1/1

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

- ○ TRUE
- ◉ FALSE ✓
- ○ Compiler Error
- ○ Output is compiler dependent

✓ 1/1

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

- ○ 0
- ◉ 1 ✓
- ○ Big negative number
- ○ -1

✓ Predict the output * 1/1

```
#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

✓ Predict the output of the below program: * 1/1

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

- ( ) (A) 112
- ( ) (B) 52
- (•) (C) 512 ✓
- ( ) (D) 0

✓ 1/1

```
#include <stdio.h>

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

    printf("%d", i);

    return 0;
}
```

- ○ 1
- ◉ 3 ✓
- ○ 2
- ○ Compile time error
- ○ Garbage value

هانت :) اضحك كده :) 37 of 37 points

✓ (Error in execution) is 1/1

- ○ Syntax error
- ◉ Runtime error ✓
- ○ Logical error
- ○ Semantic error
- ○ Linker error

✓ Which of the following lines of code can be used to set specific bit of A * 1/1

- A &= ~(1 << bit)
- (A & 1 << bit) != 0
- (●) A |= 1 << bit ✓
- A & ~B
- A ^= 1<<bit

✓ Conversion of smaller number to larger number is .....conversion.Conversion of integer type data to float.float i=0;int j=10;i=j; 1/1

- (●) Implicit ✓
- Explicit
- other

✓ int i=4 , j=7 , k ; k= j|| i++ &&1 ; *1/1

- k=7
- i=5
- i=4
- (●) k=1 ✓

✓ .....Conversion is done automatically. * 1/1

- (●) Implicit ✓
- ( ) Explicit
- ( ) other

✓ * 1/1

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

- ( ) 3
- ( ) 4
- ( ) 5
- (●) compile error ✓

✓ for(short i=0;i>=(unsigned short)0;i++); * 1/1

- (●) finite loop ✓
- ( ) infinite loop

✓ int X, i=4, j=7; X=j || i++ && 1 * 1/1

- (●) X=1 , i =4 ✓
- ( ) X=7 , i =4
- ( ) X=7 , i =5
- ( ) compile error

✓ i=4 ; j = (++i)++ * 1/1

- ( ) 6
- ( ) 5
- ( ) execution error
- (●) compilation error ✓

✓ __type__ __var__ = __value__ ; 1/1

- ( ) declaration
- (●) Initialization ✓
- ( ) definition
- ( ) strange class
- ( ) Casting

✓ Which of the following lines of code can be used to reset specific bit of A * 1/1

- (●) A &= ~(1 << bit) ✓
- ( ) (A & 1 << bit) != 0
- ( ) A |= 1 << bit
- ( ) A & ~B

✓ Conversion of larger number to smaller number is ..... conversion.float k=123.456int i= (int) k *1/1

- ( ) Implicit
- (●) Explicit ✓
- ( ) other

✓ for(int i=0;i>=0;i++); * 1/1

- (●) finite loop ✓
- ( ) infinite loop

✓ int a=b=c=1; * 1/1

- a =1 and b =1 and c=1
- (●) Compiler error ✓
- runtime error

✓ * 1/1

```
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

✓ int i=0 , j=1 , k=2 , m; m= i++ || j++ || k++; printf("%d %d %d %d", i , j , k , m); *1/1

- 1 2 3 1
- 1 2 2 1 ✓
- 1 2 2 2
- 1211

✓ a=b=c= 10; if (a==b==c); * 1/1

- true
- False ✓

✓ for(unsigned int i=0;i>=0;i++); * 1/1

- finite loop
- infinite loop ✓

✓ Error in the result of program 1/1

- ○ Syntax error
- ○ Runtime error
- ◉ Logical error ✓
- ○ Semantic error
- ○ Linker error

✓ (Error in writing syntax) is 1/1

- ◉ Syntax error ✓
- ○ Runtime error
- ○ Logical error
- ○ Semantic error
- ○ Linker error

✓ * 1/1

```c
#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

✓ unsigned short i = 0xFFFF; while (i++!=0) 1/1
printf("%d",++i);

- 0xFFFF 0x0
- infinite loop ✓
- Finite loop

✓ what is the output ...? 1/1

```c
#include "stdio.h"

void main ()
{
    int i = 0x10+010+10;
    printf ("%x",i) ; // ("i= %d ", i);
}
```

- (●) 22 ✓
- ( ) 34
- ( ) 30
- ( ) 38

✓ * 1/1

Predict the output of the following code ?

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

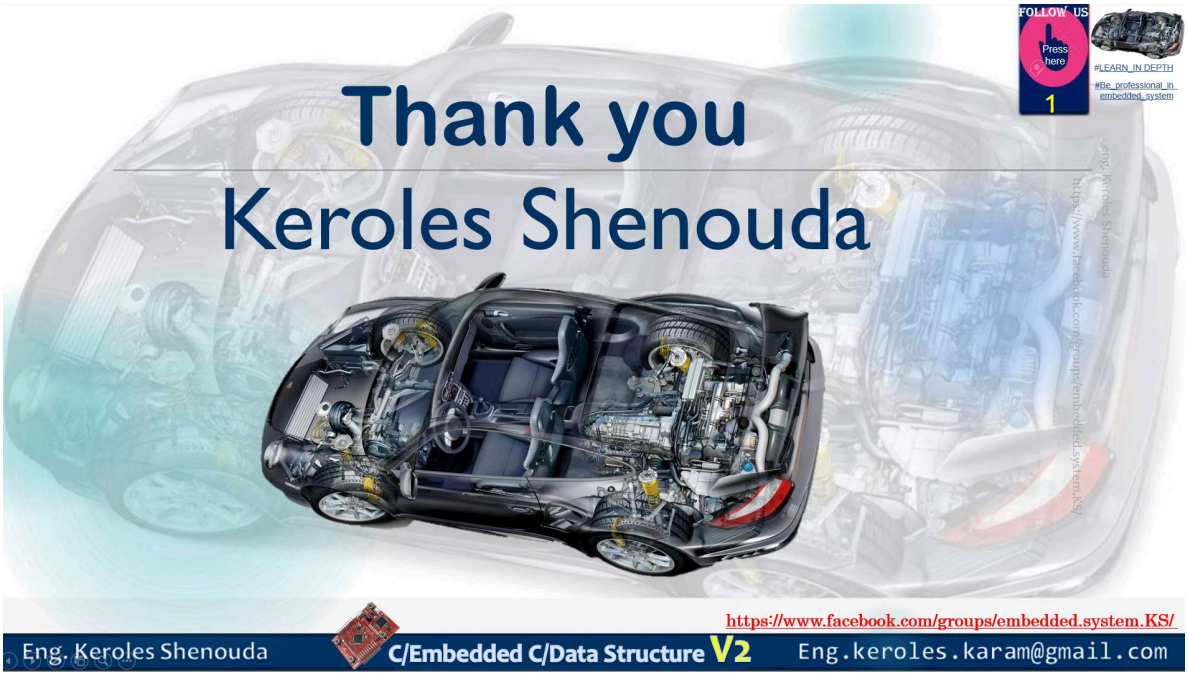
- ( ) 12 10
- ( ) 12 12
- ( ) 12 11
- (●) Output may Vary from Compiler to Compiler ✓

✓ char ch = ‘a’ = 97 ; switch (ch) { case 97: printf("97") ; break; case ‘a’: printf ("a") ; break;} *1/1

- 97
- a
- 97 a
- (●) compilation error ✓
- runtime error

✓ type casting is to * 1/1

- Implicit
- (●) Explicit ✓
- other



✓ a = 5 , b = 4 if (a==b); printf ("Equal"); 1/1

- (●) Print “Equal” ✓
- ( ) Doesn’t print

✓ int x=3; float y=3.0; 1/1
if(x==y) {printf (“True”);}

- (●) print True ✓
- ( ) will not print true

✓ int i=2 , j=5 , k=10 ; *1/1
a = i >1 ? j<1 || k<1 ? 100:200:300;

- (●) a= 200 ✓
- ( ) a = 100
- ( ) a= 300
- ( ) a = 0

✓ .....Conversion is done programatically. * 1/1

- ( ) Implicit
- (●) Explicit ✓
- ( ) other

✓ * 1/1

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

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

- (●) fff0 ✓
- ( ) fff1
- ( ) fff2
- ( ) fff3
- ( ) fff4

✓ extern __type__ __var__ ; 1/1

- (●) declaration ✓
- ( ) Initialization
- ( ) definition
- ( ) strange class
- ( ) Casting

✓ Cannot be generated the link of wrong function prototyping, or incorrect header files 1/1

- ○ Syntax error
- ○ Runtime error
- ○ Logical error
- ○ Semantic error
- ◉ Linker error ✓

✓ Which of the following lines of code can be used to toggle specific bit of A *1/1

- ○ A &= ~(1 << bit)
- ○ (A & 1 << bit) != 0
- ○ A |= 1 << bit
- ○ A & ~B
- ◉ A ^= 1<<bit ✓

✓ int i=1; if (i++&&(i==1)) * 1/1

- ○ condition true
- ◉ false ✓

✓ Write “Hello World” without semicolon * 1/1

- [ ] printf("Hello World");
- [x] if(printf("Hello World")); ✓
- [ ] if(printf("Hello_World")){}
- [ ] switch(fprintf("Hello World"));

✓ * 1/1

```
#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