

Question 1

1.1

```
int x = 7, y = 3, z = 2;
```

z: **24**

1.2

1 7 1 4

0 1

1.3

x:**3** y:**8** z:**12** w:**0**

1.4

x: **15** y: **15** z:**14** w:**28**

1.5 2.400 2.700

1.6 yes

4 2

1.7

11111111 0377 or 0XFF or 255

01111111 0177 or 0X7F or 127

1.8

59	12
<hr/>	<hr/>
invalid	invalid
<hr/>	<hr/>
invalid	23
<hr/>	<hr/>

1.9

1) **turn on bit-3, keep other bits**

2) turn off bit-5, keep other bits

3) 0 0 0 1 0 1 0 1 1 1

4) 1 1 0 0 0 0 0 0 0 0

Parenthese in 1) can be removed. Because << has higher precedence than |,

Parenthese in 4) can be removed. Because ~ has higher precedence than &,

```
int mask = 59      int mask = 0X3B      int mask = 073
```

Other slu: `int mask = 1 << 5 | 1 << 4 | 1 << 3 | 1 << 1 | 1` = 48 | 11 = 32 | 16 | 8 | 3 anything evaluate to 59

turn on bit-0, 1, 3, 4, 5, keep others or 1 1 0 1 1 1 1 1 1 1

1.10

- illegal
- 1 5 3 2 25
- 1 0 0 0 0 1 \0 \0 \0 \0 is wrong
- 1 5 3 0 0 1 5 3 \0 \0 is wrong

1.11

Hello world-12-11

Hi-12-2

1.12

Hello world-20-11

Hello worldHi-20-13

1.13 F 70

G 71

H 72

I 73

1.14

4.200

1.15

8.000

1.16

No. Index is (a local variable), not initialized to 0

1.17

Clause one

Entered 1

Clause one

Entered 1

Clause one

Entered 1

1.18

x:100 y: 200 z: 0

x:100 y: 210 z: 10

x:100 y: 220 z: 20

1.19

- a. y is static in function.c, which makes its scope to be within function.c only, and thus not accessible in main.c – “undefined reference to y”

1.20

Scanf needs to change the value of its parameter.

Since c is call/pass by value, to change/set the value of age and wage, we need the address/pointer of the variables, and thus &age, &wage.

In C, array name contains the address of the first elements arr=&arr[0] (array name is already a pointer). So we just pass it to the function without &

Question 2

2.1

100 - 8031 - 8031 - 8031

2.2 See solution of lab5

2.3

y's value is set to 69 now

x's value is set to 70 now

x's value is set to 700 now

y's value is set to 70 now

x's value is set to 680 now

x: 680 y: 70

2.4 Consider the following ANSI C program.

1001	3004	5008
1001	3004	5008
1006	3024	5048

2.5

valid	invalid
valid	invalid
valid	invalid
valid	invalid
valid	invalid
valid	invalid
valid	invalid
valid	invalid

2.6

1000 1000

1000 1000

1004 1004

1008 1008

2.7

2012 10

2016 126

2016 126

2020 22

4 1 1

17, 4, 58, 10, 126, 22, 33, 456

2.8

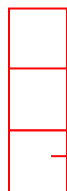
1) False

2) f. All of the above

2.9

argv is an array. Each element of the array is a pointer to char. That is, argv is an array of (char) pointers

ptr



→ char

char * ptr [3]