Question 1

1.1

```
int x = 7, y = 13, z = 2;
```

z: **64**

1.2

1 4 1 7

1 0

1.3

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

1.4

x: **5** y: **5** z:**4** w:**48**

1.5 **2.700** **2.400**

1.6 **yes**

4 **2**

1.7

01111111 0177 or 0X7F or 127

11111111 0377 or 0XFF or 255

1.8

91	invalid
invalid	**12**
23	**invalid**

1.9

1) turn on bit-**6**, keep other bits

2) turn off bit-3, keep other bits

3) **1 1 0 0 0 0 0 0 0 0**
4) 0 0 0 1 0 1 0 1 1 1

Parenthese in 1) can be removed. Because << has higher precedence than |,
Parenthese in 3) can be removed. Because ~ has higher precedence than &,

int mask = 45 int mask = 0X2D int mask = 055

Other slu: int mask = 1 <<5 | 1 << 3 | 1 <<2 | 1 =32 | 13 = 32|8|4|1 ..... anything evaluate to 45

turn on bit-0, 2, 3, 5, keep others or **1 1 0 1 1 1 1 1 1 1**

1.10

- **1 5 13 2 25**
- **Illegal**
- **1 5 13 0 0 0 1 5 13 \0 \0 \0 is wrong**
- **1 0 0 0 0 0 1 \0 \0 \0 \0 \0 is wrong**

1.11

Hello the world-16-15

OK-16-2

1.12

Hello the world-30-15

Hello the worldOK-30-17

1.13 f 102

g 103

h 104

i 105

1.14

8.000

1.15

4.200

1.16

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

1.17

Clause one

Entered 1

Clause two

Entered 2

Clause two

Entered 2

1.18

x:10 y: 0 z:200

x:10 y: 2 z: 202

x:10 y: 4 z: 204

1.19

a. x 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 x"

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

10-1099-1099-1099

2.2 See solution of lab5

2.3

y's value is set to 690 now

x 's value is set to 689 now

x 's value is set to 6890 now

y 's value is set to 689 now

x 's value is set to 6910 now

x: 6910 y: 689

2.4 Consider the following ANSI C program.

3001	5004	7008
3001	5004	7008
3005	5020	7040

2.5

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

2.6

4000 4000

4000 4000

4004 4004

4008 4008

2.7

7012 100

7016 1206

7016 1206

7020 202

4 1 1

107, **31**, 508, 100, 1206, 202, **453**, 4056

2.8 **1)** **False**

2) e. 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