

4.

- 1 无符号整型 unsigned int 数据范围  $[0, 2^{(16-1)}-1]$
- 2 有符号整型 signed 数据范围  $[-2^{(15-1)}, 2^{(15-1)}-1]$
- 3 有符号短整型 short 数据范围  $[-2^{(15-1)}, 2^{(15-1)}-1]$

数据类型	16位平台	32位平台	64位平台
char	1(个字节)	1	1
short	2	2	2
int	2	4	4
unsigned int	2	4	4
float	4	4	4
double	8	8	8
long	4	4	8
long long	8	8	8
unsigned long	4	4	8
指针	2	4	8
最大存储空间	$2^{16}$	$2^{32}$	$2^{64}$

9.

```
CS_HW > c5 > C t8.c > main()
1  #include<stdlib.h>
2  #include<stdio.h>
3
4  void main(){
5      int x=-1;
6      unsigned int uy=4294967295UL;
7      int a=(x^uy);
8      int b=(x && uy);
9      printf("unsigned x is : %u \n",x);
10     printf("signed uy is : %d \n",uy);
11     printf("a = %d \n",a);
12     printf("b = %d \n",b);
13     return;
14 }
```

```

unsigned x is : 4294967295
signed uy is : -1
a = 0
b = 1

```

将 x 作为无符号整数来输出的话，-1 的第一位符号位被改变，就变成了 4294967295，uy 作为整数输出也同理变成了 -1。

逻辑与的汇编实现：(X && uy) 逻辑就是判断两个数是否为 0；如果其中一个数为 0，返回 0；都不为 0，返回 1。汇编还是用了分支结构跳转来实现，通过与 0 比较，实现了最好情况只用比较一次的方法，避免重复设置符号位 Z 之后再与 1 相与，最后用 0 扩展把结果放回到 32 位寄存器里面。

11.

Sign	Exp	Frac							
0	1	1	1	1	1	1	1	1	1

在不算正无穷的情况下：

8 位采用偏置表示，偏置一般为  $2^8 - 1 - 1 = 127$ 。指数全 0 和全 1 分别保留用于非正规数和特殊值（如无穷大、NaN），因此正规数的指数编码范围是 1 到 254。

因此，最大正规数为 +1.5 的 127 次方。

17.

A【10111110】=B【10011111】

19.

1. 正数中最小的非规格化数 0 0000 0001
2. 正数中最大的非规格化数 0 0000 1111
3. 正数中最小的规格化数 0 0001 0000
4. 正数中最大的规格化数 0 1110 1111