1. 选择题

（13期中）1、变量 x 的值为 0x01234567，地址 &x 为 0x100；则该变量的值在 x86 和 Sun 机器内存 中的存储排列顺序正确的是 ( )

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 选项 | 机器类型 | 地址 | | | |
| 0x100 | 0x101 | 0x102 | 0x103 |
| A | x86 | 67 | 45 | 23 | 01 |
| Sun | 01 | 23 | 45 | 67 |
| B | x86 | 76 | 54 | 32 | 10 |
| Sun | 01 | 23 | 45 | 67 |
| C | x86 | 01 | 23 | 45 | 67 |
| Sun | 67 | 45 | 23 | 01 |
| D | x86 | 01 | 23 | 45 | 67 |
| Sun | 01 | 23 | 45 | 67 |

（13期中）2 、假设下列 int 和 unsigned 数均为 32 位， int x = 0x80000000;

unsigned y = 0x00000001;

int z = 0x80000001;

以下表达式正确的是 ( )

A. (-x) < [0](#_bookmark1)

B. (- 1) > y

C. (z<<3) == (z\*8)

D. y\*24 == z<<5 - z<<3

（14期中）3.假设下列 unsigned 和 int 数均为 32 位， unsigned x = 0x00000001;

int y = 0x80000000;

int z = 0x80000001;

以下表达式正确的是

A. (-1) < x

B. (- y) > -1

C. ~ y + y == -1

D. (z<<4) > (z\*16)

（14期中）4.下面表达式中为“真”的是：

A．(unsigned) -1 < -2

B．2147483647 > (int) 2147483648U

C. (0x80005942 >> 4) == 0x09005942

D．2147483647 + 1 != 2147483648

（15期中）5. 以下说法错误的是

A. 负数加上负数结果可能为正数

B. 正数加上正数结果可能为负数

C. 用&和~可以表示所有的逻辑与或非操作

D. 用&和|可以表示所有的逻辑与或非操作

（17期中）6. 假定一个特殊设计的计算机，将 int 型数据的长度从 4-Byte 扩展为 4N-Byte，采用大端法(Big Endian)。现将该 int 型所能表示的最小负数 写入内存中，如下图所示。其中每个小矩形代表一个 Byte，请问 X 位置这个 Byte 中的值是多少？



A. 000000002 B. 011111112 C. 100000002 D. 111111112

（18期中）7. 在采用小端法存储机器上运行下面的代码，输出的结果将会是？

(int，unsigned 为 32 位长，short 为 16 位长，0~9 的 ASCII 码分别是 0x30~0x39)

char \*s = "2018";

int \*p1 = (int \*)s;

short s1 = (\*p1)>>12;

unsigned u1 = (unsigned) s1;

printf("0x%x\n",u1);

1. 0x00002303 B) 0x00032303 C) 0xffff8313 D) 0x00008313

（18期中）8.考虑如下函数

void XOR(int x, int y) {

y = x ^ y;

x = x ^ y;

y = x ^ y;

printf(“%d %d”,x, y);

}

XOR(a,b)的输出结果为？

A) a, b B) b, a C) b,0 D) b, a^b

1. 填空与简答题

（13期中）1. 判断下表中每一行表达式对或错。如果错， 请举出反例或简要说明原因

int x, y;

unsigned u, v;

|  |  |  |
| --- | --- | --- |
|  | True or false | 原因或举出反例 |
| if x<0, then x\*2 <0 |  |  |
| u<= - 1 |  |  |
| if x> y, then -x < -y |  |  |
| if u> v, then -u > -v |  |  |

（14期中）2.假设下列 unsigned 和 int 数均为 5 位(有符号整型用补码运算表示)，在下 表中填入正确答案

int y = - 7;

unsigned z = y;

|  |  |  |
| --- | --- | --- |
|  | Decimal  Representation | Binary Representation |
| z |  |  |
| y – z |  |  |
| TMin |  |  |

（15期中）3.对于下面的每一个表达式， 请选择以下选项中的一个或多个(即“不定项”)，使得该表达式恒成立， 如果没有满足条件的选项则选 E。

A. < B. > C. == D. != E. none

题目中出现的变量定义如下:

int x, y;

unsigned ux = x;

1) 如果 x > 0, 则 x + 1 0

2) 如果 x > y, 则 ux y

3) 如果 ((x << 31) >> 31) < 0, 则 x & 1 0

4) 如果 ((unsigned char)x >> 1) < 64， 则 (char)x 0

5) xˆyˆ(~x) - y yˆxˆ(~y)-x

6) (((!!ux)) << 31) >> 31) (((!!x) << 31) >> 31)

（18期中）4.生成任意 int 类型的 x，然后将它转换为 unsigned 类型：

int x = random();

unsigned ux = (unsigned) x;

对于以下每个 C 语言表达式， 你需要判断它的值是否恒为 1。如果是，圈出 Y； 否则圈出 N。

|  |  |
| --- | --- |
| 表达式 | 恒为 1？ |
| (x >= 0) || (x < ux) | Y N |
| ( (x>>1) << 1) <= x | Y N |

1. 答案与解析：

选择：

1. A

考察大端、小端；同时 sun 是大端、 x86 是小端，大端法低地址是高位，小端法低地址是低位

1. ABCD；考虑到运算符的优先顺序，选 ABC 也算对
2. int 中 0x80000000 的相反数还是自己

B. signed (- 1)和unsinged y比较，都按照 unsigned，所以强制类型转换后(-1)很大

C: unsigned, signed 左移三位 = \*8

D：应该是相等关系；signed左移之后和unsigned y\*24相比都看成 unsigned（但要注意位运算优先级！）

3.C

A 错误 signed (-1) 和 unsigned x 比较， 都按照 unsigned，所以强制类型转换后( -1)很大

B 错误 int 中 0x80000000 的相反数还是自己，是最小的负数

C 正确 相等关系； ~y + y == 0xFFFFFFFF 是 -1 的补码表示 （-x=~x+1）

D 错误 应该是相等关系 unsigned, signed左移 4 位 相当于 \*16

4.B

A 全都按照unsigned比较

B 正确，强制转化成int后右侧为负数

C 默认为有符号使用算术右移

D 使用2147483648这样的数字会被默认成long或者unsigned，这样左侧不会导致溢出

1. D

无法用&和|表示逻辑非

6. C

最小的负数为 1000000… ，大端法将最高位放到最低地址

7. C

本题考查大小端存储， 以及整数类型转化。字符串在存储时不区分大小端， 前面的字符存储在低地址，而在被转化成整型的时候低地址被视为低位，因此\*p1 为 0x38313032，s1 为 0x8313。在由 short 转化为 unsigned 的时候， 我们要先改变大 小，之后完成有符号到无符号的转换，因此 u1 为 0xffff8313

1. B

第一步：x1=x0,y1=x0^y0

第二步：x2=x1^y1=x0^(x0^y0)=y0,y2=y1

第三步：x3=x2=y0,y3=x2^y2=y0^(x0^y0)=x0

填空与解答：

1.

|  |  |  |
| --- | --- | --- |
|  | True or false | 原因或举出反例 |
| if x<0, then x\*2 <0 | F | X= -2w- 1 |
| u<= - 1 | T | - 1 作为无符号来比大于 u |
| if x> y, then -x < -y | F | X=0, y= -2w- 1 |
| if u> v, then -u > -v | F | U=2, v=1 |

2.

|  |  |  |
| --- | --- | --- |
|  | Decimal  Representation | Binary Representation |
| z | 25 | 1 1001 |
| y – z | 0 | 0 0000 |
| TMin | -16 | 1 0000 |

3.

1. D 只能保证x+1!=0，可能大于0，也可能小于0（溢出）
2. D x>y可能是因为正负，但转成unsigned之后负数可能很大
3. B,D (x<<31)>>31实际上提取了x最低一位，其小于0说明最低一位是1，于是&1为1>0
4. E 显然可能为0也可能不为0，只能选E
5. C 首先我们要关注运算顺序，减法优先于位异或，于是表达式实际上是x^y^((~x)-y)与x^y^((~y)-x)，而-x=~x+1，于是~x-y=~y-x=-x-y-1，于是为==
6. C x为0是自然的，x不为0时!返回值为有符号型。

4.

|  |  |
| --- | --- |
| 表达式 | 恒为 1？ |
| (x >= 0) || (x < ux) | N: e.g. x = -1. |
| ( (x>>1) << 1) <= x | Y: x>>1 rounds toward minus infinity. |