**第二章课后习题**

**2.55**

运行结果如下：



又int、float运行结果可知，此机器为小端序。

**2.59**

设最初两个数为x、y，生成的数z的最低有效位来自x，其余位来自y，则有

z = (x & 0x000000FF) || (y & 0xFFFFFF00)

**2.63**

1. #include <stdio.h>
3. **int** w;
4. unsigned srl(unsigned x,**int** k){
5. /\* Perform shift arithmettically \*/
6. unsigned xsra = (**int**) x >> k;
7. **int** le\_srl = 0;
8. **for**(**int** i = 0;i <= 31-k;i++){
9. le\_srl <<= 1;
10. le\_srl += 1;
11. }
12. xsra = xsra & (unsigned)le\_srl;
13. **return** xsra;
14. }
16. **int** sra(**int** x,**int** k){
17. **int** xsrl = (unsigned) x >> k;
18. **int** le = 0;
19. unsigned le\_sra;
20. **if**(x < 0){
21. **for**(**int** i = 0;i <= 31-k;i++){
22. le \*= 2;
23. le += 1;
24. }
25. le\_sra = (unsigned)le ^ 0xFFFFFFFF;
26. xsrl = (**int**)le\_sra ^ xsrl;
27. }
28. **return** xsrl;
29. }
31. **int** main(){
33. w = 8 \* **sizeof**(**int**);
34. printf("int的位数为%d\n",w);
36. unsigned x = 0x80000002;
37. unsigned y = srl(x,1);
38. printf("y = %x\n",y);
40. **int** a = 0x80000002;
41. **int** b = sra(a,1);
42. printf("b = %x\n",b);
44. **return** 0;
45. }

**2.67**

A. int 占4个字节，共32位。在

Int beyond\_msb = 1 << 32;

中，移位数量超出了字长，在sun spark上并不会进行取模操作，C语言对此也没有特别规定，故产生警告。

B.

1. **int** bad\_int\_size\_is\_32(){
2. **int** i = 1 ;
3. **int** set\_msb = 1 << 31;
4. **int** beyond\_msb = 1 << 31 << 1;
5. printf("set\_msb = %d\n",set\_msb);
6. printf("beyond\_msb = %.8x \n",beyond\_msb);
7. **return** set\_msb && !beyond\_msb;
8. }

C.

1. **int** bad\_int\_size\_is\_32(){
2. **int** i = 1;
3. **int** set\_msb = 1 << 15 << 15 << 1;
4. **int** beyond\_msb = 1 << 15 << 15 << 2;
5. printf("set\_msb = %d\n",set\_msb);
6. printf("beyond\_msb = %.8x \n",beyond\_msb);
7. **return** set\_msb && 7!beyond\_msb;
8. }

**2.71**

（1）如果抽出的是负数，这段代码返回的是无符号数。

（2）

1. **int** xbyte(packed\_t word,**int** bytenum){
2. **return** ((**int**)word <<(24-(bytenum << 3)))>>24;
3. }

**2.75**

1. #include <stdio.h>
3. **int** w;
5. unsigned unsigned\_high\_prod(unsigned x,unsigned y){
6. **int** W = signed\_high\_prod(x,y);
7. **int** sighed\_x = x >> (w-1);
8. **int** sighed\_y = y >> (w-1);
9. **return** W + x \* sighed\_y + y \* sighed\_x;
10. }

13. **int** main(){
14. **int** x = 0x12345678;
15. **int** y = 0x12345678;
16. w = 32;
17. printf("%d\n",unsigned\_high\_prod(x,y));
18. **return** 0;
19. }

**2.79**

1. #include <stdio.h>
3. **int** threefourths(**int** x){
4. **int** x\_div2 = x >> 1;
5. **int** x\_div4 = x >> 2;
6. **int** x\_mul2 = x << 1;
7. **int** x\_last = ((x \* 0x0011) + (x\_mul2 \* 0x0011))>>2;
8. **return** x\_div2 + x\_div4 + x\_last;
9. }

12. **int** main(){
13. **int** x = 0x7845678;
14. printf("%d\n",threefourths(x));
15. **return** 0;
16. }

**2.83**

(1) x = Y/(2 \*\*k - 1)

(2) (a) 5/7

(b) 2/5

(c) 19/63

**2.87**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 描述 | Hex | M | E | V | D |
| -0 | **0x8000** | **0** | **-14** | -0 | -0.0 |
| 最小的>2的值 | **0x4001** | **1025/1024** | **1** | **1025\*2^(-9)** | **2.001953** |
| 512 | **0x6000** | **1** | **9** | 512 | 512.0 |
| 最大的非规格化束 | **0x03FF** | **1023/1024** | **-14** | **1023\*2^(-24)** | **0.000061** |
| -∞ | **0xFC00** | --- | --- | -∞ | -∞ |
| 3BB0 | 3BB0 | **123/64** | **-1** | **123\*2^(-7)** | **0.960938** |

**2.91**

(1) 3.141593

(2) 11.001001001……

(3) 第9位

**2.95**

1. **typedef** unsigned float\_bits;
2. float\_bits float\_half(float\_bits f){
3. unsigned sign = f & 0x80000000;
4. unsigned exp = (f>>23) & 0xff;
5. unsigned flac = f & 0x7fffff;
6. unsigned flag = f & 0x03;
7. unsigned usd\_f = f & 0x7fffffff;
8. **if**(exp == 0xff){
9. **return** f;
10. }**else** **if**(exp == 0)
11. {
12. /\* code \*/
13. flac = flac >> 1;
14. **if**(flag == 0x3){
15. flac += 1;
16. }
17. }**else** **if**(exp == 1){
18. usd\_f >> 1;
19. **if**(flag == 0x3){
20. usd\_f += 1;
21. }
22. exp = usd\_f >> 23;
23. flac = usd\_f & 0x7fffff;
24. }**else**{
25. exp = exp - 1;
26. }
27. **return** sign << 31 | exp << 23 | flac;
28. }