

Chun-Wei Chen  
CSE 351  
Homework 1  
04/19/13

1.

- A.  $(x \ll 4) + x$
- B.  $x - (x \ll 3)$
- C.  $(x \ll 6) - (x \ll 2)$
- D.  $(x \ll 4) - (x \ll 7)$

2.

- A. Always yield 1. float  $\rightarrow$  int the number may be rounded but not overflow; on the other hand, since int  $\rightarrow$  double is exact conversion, double  $\rightarrow$  float will be essentially the same as initial int value to float.
- B. If  $x = 0x7FFFFFFF$ ,  $y = \sim x + 1$ , then  $dx - dy \neq (\text{double}) (x - y)$  since  $x - y$  will overflow
- C. Always yield 1. The result of the sum of three int values will at most take 34 bits to store, and double has 52 bits of significand (mantissa), which means either left-hand side or right-hand side of the expression won't encounter overflow; therefore, it'll always yield 1.
- D.  $x = 0x7FFFFFFF$ ,  $y = x$ ,  $z = 9999$ . double only has 52 bits of significand but the exact result of the multiplication of these three values will need more than 52 bits to store, so the result will be rounded and no guarantee to get the same result on left-hand side and right-hand side of the expression.
- E. If  $x = 0$  or  $z \neq 0$ , the expression won't yield 1.

3.

The problem is float and double don't have the same bits of mantissa, so calling `test(1/3., 1/3.)` or `test(0x7FFFFFFF, 0x7FFFFFFF)` will return 0.

4.

$15.0 = 7.5 * 2 = 3.75 * 2^2 = 1.875 * 2^3$   
So  $M = 1.875$ ,  $E = 3$ ,  $\text{frac} = 1.875 - 1 = 0.875$

5.

- A. symbol's name
- B. `sar` and `sal`
- C.

Show dependency info of `btest` and info related to symbol version etc.

`bash-4.2$ ldd -v btest`

```
linux-vdso.so.1 => (0x00007fff5a711000)
libm.so.6 => /lib64/libm.so.6 (0x0000003679c00000)
libc.so.6 => /lib64/libc.so.6 (0x0000003679800000)
/lib64/ld-linux-x86-64.so.2 (0x0000003679000000)
```

Version information:

`./btest:`

```
libc.so.6 (GLIBC_2.2.5) => /lib64/libc.so.6
```

```

/lib64/libm.so.6:
    libc.so.6 (GLIBC_PRIVATE) => /lib64/libc.so.6
    libc.so.6 (GLIBC_2.2.5) => /lib64/libc.so.6
/lib64/libc.so.6:
    ld-linux-x86-64.so.2 (GLIBC_2.3) => /lib64/ld-linux-x86-64.so.2
    ld-linux-x86-64.so.2 (GLIBC_PRIVATE) => /lib64/ld-linux-x86-64.so.2

```

Summary information from the section headers of btest  
 bash-4.2\$ objdump -h btest

btest: file format elf64-x86-64

Sections:

| Idx  | Name                                  | Size     | VMA              | LMA              | File off         | Algn     |
|------|---------------------------------------|----------|------------------|------------------|------------------|----------|
| 0    | .interp                               | 0000001c | 0000000000400200 | 0000000000400200 | 0000000000400200 | 00000200 |
| 2**0 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 1    | .note.ABI-tag                         | 00000020 | 000000000040021c | 000000000040021c | 000000000040021c | 0000021c |
| 2**2 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 2    | .note.gnu.build-id                    | 00000024 | 000000000040023c | 000000000040023c | 000000000040023c | 0000023c |
| 2**2 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 3    | .gnu.hash                             | 00000024 | 0000000000400260 | 0000000000400260 | 0000000000400260 | 00000260 |
| 2**3 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 4    | .dynsym                               | 00000228 | 0000000000400288 | 0000000000400288 | 0000000000400288 | 00000288 |
| 2**3 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 5    | .dynstr                               | 00000110 | 00000000004004b0 | 00000000004004b0 | 00000000004004b0 | 000004b0 |
| 2**0 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 6    | .gnu.version                          | 0000002e | 00000000004005c0 | 00000000004005c0 | 00000000004005c0 | 000005c0 |
| 2**1 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 7    | .gnu.version_r                        | 00000020 | 00000000004005f0 | 00000000004005f0 | 00000000004005f0 | 000005f0 |
| 2**3 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 8    | .rela.dyn                             | 00000030 | 0000000000400610 | 0000000000400610 | 0000000000400610 | 00000610 |
| 2**3 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |
| 9    | .rela.plt                             | 00000198 | 0000000000400640 | 0000000000400640 | 0000000000400640 | 00000640 |
| 2**3 |                                       |          |                  |                  |                  |          |
|      | CONTENTS, ALLOC, LOAD, READONLY, DATA |          |                  |                  |                  |          |

```

10 .init      0000000e 00000000004007d8 00000000004007d8 000007d8
2**2
      CONTENTS, ALLOC, LOAD, READONLY, CODE
11 .plt       00000120 00000000004007f0 00000000004007f0 000007f0 2**4
      CONTENTS, ALLOC, LOAD, READONLY, CODE
12 .text      00000eb4 0000000000400910 0000000000400910 00000910
2**4
      CONTENTS, ALLOC, LOAD, READONLY, CODE
13 .fini      00000009 00000000004017c4 00000000004017c4 000017c4 2**2
      CONTENTS, ALLOC, LOAD, READONLY, CODE
14 .rodata    000006e5 00000000004017d0 00000000004017d0 000017d0
2**3
      CONTENTS, ALLOC, LOAD, READONLY, DATA
15 .eh_frame_hdr 00000114 0000000000401eb8 0000000000401eb8
00001eb8 2**2
      CONTENTS, ALLOC, LOAD, READONLY, DATA
16 .eh_frame   000003bc 0000000000401fd0 0000000000401fd0 00001fd0
2**3
      CONTENTS, ALLOC, LOAD, READONLY, DATA
17 .init_array 00000008 0000000000602390 0000000000602390 00002390
2**3
      CONTENTS, ALLOC, LOAD, DATA
18 .fini_array 00000008 0000000000602398 0000000000602398 00002398
2**3
      CONTENTS, ALLOC, LOAD, DATA
19 .jcr        00000008 00000000006023a0 00000000006023a0 000023a0
2**3
      CONTENTS, ALLOC, LOAD, DATA
20 .dynamic    000001e0 00000000006023a8 00000000006023a8 000023a8
2**3
      CONTENTS, ALLOC, LOAD, DATA
21 .got        00000008 0000000000602588 0000000000602588 00002588
2**3
      CONTENTS, ALLOC, LOAD, DATA
22 .got.plt    000000a0 0000000000602590 0000000000602590 00002590
2**3
      CONTENTS, ALLOC, LOAD, DATA
23 .data       00000380 0000000000602640 0000000000602640 00002640
2**5
      CONTENTS, ALLOC, LOAD, DATA
24 .bss        04a630a8 00000000006029c0 00000000006029c0 000029c0
2**5
      ALLOC

```

```

25 .comment 0000002c 0000000000000000 0000000000000000 000029c0
2**0
    CONTENTS, READONLY
26 .debug_aranges 000000b0 0000000000000000 0000000000000000
000029ec 2**0
    CONTENTS, READONLY, DEBUGGING
27 .debug_info 00002549 0000000000000000 0000000000000000 00002a9c
2**0
    CONTENTS, READONLY, DEBUGGING
28 .debug_abbrev 0000074a 0000000000000000 0000000000000000
00004fe5 2**0
    CONTENTS, READONLY, DEBUGGING
29 .debug_line 0000056b 0000000000000000 0000000000000000 0000572f
2**0
    CONTENTS, READONLY, DEBUGGING
30 .debug_str 000007b9 0000000000000000 0000000000000000 00005c9a
2**0
    CONTENTS, READONLY, DEBUGGING
31 .debug_loc 00001d46 0000000000000000 0000000000000000 00006453
2**0
    CONTENTS, READONLY, DEBUGGING
32 .debug_ranges 00000060 0000000000000000 0000000000000000
00008199 2**0
    CONTENTS, READONLY, DEBUGGING

```

Displays the contents of btest's unwind section

```
bash-4.2$ readelf -u btest
```

The decoding of unwind sections for machine type Advanced Micro Devices X86-64 is not currently supported.

Sort symbols numerically by their address

```

bash-4.2$ nm -v btest
w _ITM_deregisterTMCloneTable
w _ITM_registerTMCloneTable
w _Jv_RegisterClasses
w __gmon_start__
U __libc_start_main@@GLIBC_2.2.5
U __sigsetjmp@@GLIBC_2.2.5
U __strdup@@GLIBC_2.2.5
U alarm@@GLIBC_2.2.5
U exit@@GLIBC_2.2.5
U getopt@@GLIBC_2.2.5
U perror@@GLIBC_2.2.5

```

U printf@@GLIBC\_2.2.5  
U puts@@GLIBC\_2.2.5  
U rand@@GLIBC\_2.2.5  
U sigaction@@GLIBC\_2.2.5  
U sigemptyset@@GLIBC\_2.2.5  
U siglongjmp@@GLIBC\_2.2.5  
U strcmp@@GLIBC\_2.2.5  
U strtof@@GLIBC\_2.2.5  
U strtol@@GLIBC\_2.2.5  
U strtoll@@GLIBC\_2.2.5  
00000000004007d8 T \_init  
0000000000400910 T \_start  
000000000040093c t call\_gmon\_start  
0000000000400960 t deregister\_tm\_clones  
0000000000400990 t register\_tm\_clones  
00000000004009d0 t \_\_do\_global\_ctors\_aux  
00000000004009f0 t frame\_dummy  
0000000000400a1c T bitAnd  
0000000000400a21 T bitXor  
0000000000400a2e T thirdBits  
0000000000400a34 T fitsBits  
0000000000400a4a T sign  
0000000000400a58 T getByte  
0000000000400a66 T logicalShift  
0000000000400a78 T addOK  
0000000000400a94 T bang  
0000000000400aa1 T conditional  
0000000000400ab6 T isPower2  
0000000000400ae0 t get\_num\_val  
0000000000400bb4 t usage  
0000000000400c24 t test\_function  
00000000004012fb T timeout\_handler  
000000000040130e T Signal  
0000000000401365 T main  
0000000000401630 T u2f  
000000000040163b T f2u  
0000000000401646 T test\_bitAnd  
000000000040164b T test\_bitXor  
0000000000401650 T test\_thirdBits  
000000000040166f T test\_fitsBits  
0000000000401690 T test\_sign  
00000000004016a3 T test\_getByte  
00000000004016c5 T test\_logicalShift  
00000000004016cc T test\_addOK

```

00000000004016e2 T test_bang
00000000004016eb T test_conditional
00000000004016f3 T test_isPower2
0000000000401730 T __libc_csu_init
00000000004017c0 T __libc_csu_fini
00000000004017c4 T _fini
00000000004017d0 R _IO_stdin_used
00000000004017d8 R __dso_handle
0000000000402388 r __FRAME_END__
0000000000602390 t __frame_dummy_init_array_entry
0000000000602390 t __init_array_start
0000000000602398 t __do_global_ctors_aux_fini_array_entry
0000000000602398 t __init_array_end
00000000006023a0 d __JCR_END__
00000000006023a0 d __JCR_LIST__
00000000006023a8 d _DYNAMIC
0000000000602590 d _GLOBAL_OFFSET_TABLE_
0000000000602640 D __data_start
0000000000602640 W data_start
0000000000602644 d timeout_limit
0000000000602660 D test_set
00000000006029c0 D __TMC_END__
00000000006029c0 A __bss_start
00000000006029c0 A _edata
00000000006029c0 B optarg@@GLIBC_2.2.5
00000000006029c8 b completed.6108
00000000006029e0 b grade
00000000006029e8 b test_fname
00000000006029f0 b global_rating
00000000006029f4 b argval
0000000000602a00 b has_arg
0000000000602a20 b arg_test_vals.5233
00000000050659a0 B envbuf
0000000005065a68 A _end

```

D.