15-213,20xx 秋季
数据实验室:操纵位分配:8 月
30 日,截止日期:9 月 12 日,星期三,晚上 11:59

Harry Bovik (bovik@cs.cmu.edu) 是这项任务的带头人。

1 简介

此作业的目的是更加熟悉整数和浮点数的位级表示。您将通过解决一系列编程“难题”来做到这一点。这些谜题中有许多都是非常人为的,但您会发现自己在解决这些谜题的过程中会更多地思考细节。

2 物流

这是一个单独的项目。所有的递交都是电子的。澄清和更正将张贴在课程网页上。

3 讲义说明

SITE-SPECIFIC:在此处插入一段说明教师将如何向学生分发datalab-handout.tar文件。

首先将 datalab-handout.tar 复制到您计划进行工作的 Linux 机器上的（受保护的)目录中。然后给出命令

unix> tar xvf datalab-handout.tar。

这将导致许多文件在目录中被解压。您将要修改和上交的唯一文件是 bits.c。

bits.c 文件包含 13 个编程难题中每一个的框架。你的任务是仅使用整数谜题的直线代码（即没有循环或条件）和有限数量的 C 算术和逻辑运算符来完成每个函数框架。具体来说，您只能使用以下八个运算符：

！ ~ & ^ | + << >>

一些函数进一步限制了这个列表。此外，您不得使用任何超过 8 位的常量。有关详细规则和所需编码风格的讨论，请参阅 bits.c 中的注释。

4 谜题

本节描述您将在 bits.c 中解决的难题。

表 1 按难度从易到难的粗略顺序列出了谜题。 “Rating”字段给出了拼图的难度等级（点数），“Max ops”字段给出了允许用于实现每个功能的运算符的最大数量。有关函数所需行为的更多详细信息，请参阅 bits.c 中的注释。您也可以参考 tests.c 中的测试函数。这些被用作参考函数来表达函数的正确行为，尽管它们不满足函数的编码规则。

姓名	说明 bitXor(x,y)	评级最大操作 1 14	
x || y 仅使用 & 和 ~。			
	最小的补码整数 tmin() isTmax(x)	1个	4个
	仅当 xx 是最大的二进制数时才为真。整数。	1个	10
	仅当 x 中的所有奇数位都设置为 1 时才为真。allOddBits(x)	2个	12
	使用 - 运算符返回 -x。取反（x）	2个	5个
是 AsciDigit(x)	如果 0x30 ≤ x ≤ 则为真。	3个	15
isLessOrEqual(x, y)	与 x 相同？ y : z conditional	3个	16
如果 x ≤ y 则为真，否则为假计算 !x 而不使用 !操作员。 logicalNeg(x))		3个	24
	howManyBits(x)	4个	12
	分钟不。在两个比较中表示 x 的位数。	4个	90后
floatFloat2Int(uf) 返回位级	返回位级等价物。 2*f 用于 fp arg。 F。 floatScale2(uf)	4个	30
等值。 fp arg 的 (int)f。 F。浮动功率 2(x)		4个	30
	返回位级等价物。 2.0 x 的整数 x。	4个	30

表 1:Datalab 难题。对于浮点数谜题，值 f 是与无符号整数 uf 具有相同位表示的浮点数。

对于浮点数谜题，您将实现一些常见的单精度浮点运算。
对于这些谜题，您可以使用标准控制结构（条件、循环），并且您可以使用 int 和无符号数据类型，包括任意无符号和整数常量。您不得使用任何联合、结构或数组。最重要的是，您不能使用任何浮点数据类型、操作或常量。相反，任何浮点操作数都将作为具有类型的传递给函数

unsigned,任何返回的浮点值都将是 unsigned 类型。您的代码应执行实现指定浮点运算的位操作。

附带的程序 fshow 可帮助您了解浮点数的结构。要编译 fshow,请切换到讲义目录并键入:

```
unix>使
```

您可以使用 fshow 查看任意模式表示为浮点数的内容:

```
unix> ./fshow 2080374784

浮点值 2.658455992e+36 位表示 0x7c000000,符号 = 0,指数 = f8,
小数 = 000000 归一化。 1.0000000000 X 2^(121)
```

您还可以给 fshow 十六进制和浮点值,它会破译它们的位结构。

5 评价

您的分数将根据以下分布从最多 67 分中计算得出:

36 个正确点。

26个性能点。

5风格点。

正确点。您必须解决的难题的难度等级介于 1 和 4 之间,因此它们的加权总和为 36。我们将使用 btest 程序评估您的函数,这将在下一节中介绍。如果谜题通过了 btest 执行的所有测试,您将获得完整的分数,否则没有分数。

性能点。在课程的这一点上,我们主要关心的是您能否得到正确的答案。
但是,我们希望向您灌输一种让事情尽可能简短的感觉。此外,有些谜题可以通过蛮力解决,但我们希望您更聪明。因此,对于每个函数,我们已经建立了允许您对每个函数使用的最大运算符数。这个限制非常慷慨,仅用于捕捉极其低效的解决方案。对于满足运算符限制的每个正确函数,您将获得两分。

风格点。最后,我们保留了 5 分,用于对您的解决方案和评论的风格进行主观评价。您的解决方案应尽可能简洁明了。您的评论应该提供信息,但不必面面俱到。

自动评分你的工作

我们在讲义目录中包含了一些自动评分工具 btest、dlc 和 driver.pl 以帮助您检查作业的正确性。

- btest：该程序检查bits.c 中函数的功能正确性。建立和
  使用它，键入以下两个命令：

  ```
  unix>使
  unix> ./btest
  ```

  请注意，每次修改 bits.c 文件时都必须重建 btest。

  您会发现一次完成一个函数并在进行时测试每个函数很有帮助。您可以使用 -f 标志指示 btest 仅测试单个函数：

  ```
  unix> ./btest -f bitXor
  ```

  您可以使用选项标志 -1、-2 和 -3 为它提供特定的函数参数：

  ```
  unix> ./btest -f bitXor -1 4 -2 5
  ```

  检查文件 README 以获取有关运行 btest 程序的文档。

- dlc：这是来自 MIT CILK 组的 ANSI C 编译器的修改版本，您可以使用
  检查是否符合每个谜题的编码规则。典型的用法是：

  ```
  unix> ./dlc 位.c
  ```

  该程序会静默运行，除非它检测到问题，例如整数谜题中的非法运算符、运算符过多或非直线代码。使用 -e 开关运行：

  ```
  unix> ./dlc -e bits.c
  ```

  使 dlc 打印每个函数使用的运算符的数量。键入 ./dlc -help 以获得命令行选项列表。

- driver.pl：这是一个驱动程序，使用 btest 和 dlc 来计算正确性和
  您的解决方案的性能点。它不需要参数：

  ```
  unix> ./driver.pl
  ```

  您的讲师将使用 driver.pl 来评估您的解决方案。

6 递交说明

SITE-SPECIFIC:在此处插入文本,告诉每个学生如何提交他们的bits.c
您学校的解决方案文件。

7 忠告

- 不要在 bits.c 文件中包含 <stdio.h> 头文件,因为它会混淆 dlc 并导致一些非直观的错误消息。您仍然可以在 bits.c 文件中使用 printf 进行调试,而无需包含 <stdio.h> 标头,尽管 gcc 会打印一条您可以忽略的警告。

- dlc 程序强制执行比 C++ 或 gcc 强制执行的更严格形式的 C 声明。特别是,任何声明都必须出现在任何不是声明的语句之前的块中（您用大括号括起来的内容）。例如,它会抱怨以下代码:

```
int foo(int x) {
    int a = x;一个 * = 3;
                        /* 不是声明的语句 */ int b = a; /* 错误:此处不允许声明 */
}
```

8 “打败教授”竞赛

为了好玩,我们提供了一个可选的“打败教授”竞赛,让您可以与其他学生和教师竞争,以开发出最有效的谜题。目标是使用最少数量的操作员解决每个 Data Lab 难题。每个拼图匹配或击败讲师操作员计数的学生就是赢家!

要提交您的参赛作品,请输入:

```
unix> ./driver.pl -u       你的昵称
```

昵称不得超过 35 个字符,可以包含字母数字、撇号、逗号、句点、破折号、下划线和符号。您可以随时提交。您最近提交的内容将出现在实时记分板上,仅由您的昵称标识。您可以通过将浏览器指向

```
http://$SERVER_NAME:$REQUESTD_PORT
```

SITE-SPECIFIC:将$SERVER_NAME和$REQUESTD_PORT替换为您在./contest/Contest.pm文件中设置的值。