

Datalab 2023

实验目的和内容

- 实验目的：深入理解 `int` 和 `float` 数据的存储方式和位操作
- 实验内容：按照要求实现22个函数（C语言，不支持C11及更新标准）
- DDL：10月18日23：59

实验环境

1. 实验代码在线评测平台为 `ics.men.ci`。
2. 本实验需要在Linux环境下进行。我们为同学们提供了Linux环境的服务器 `ics.ruc.rvalue.moe`，用于获取实验文件并编写代码。

实验步骤

1. 使用SSH登录服务器 `ics.ruc.rvalue.moe`，**初次登录时必须修改密码。（禁止弱密码）**
2. 将 `/mnt/ics1-2020` 目录下的 `datalab-handout.tar` 复制到你的用户文件夹，在你的用户文件夹下使用 `tar -xvf datalab-handout.tar` 解压。
3. 在 `bits.c` 中仔细阅读代码开头的说明。
4. 在 `bits.c` 中实现函数。
5. 登录评测平台 `ics.men.ci`，**初次登录时必须修改密码。（禁止弱密码）**
6. 在datalab比赛中提交你的代码，然后向着排行榜上更高的位置进发！

在本地检查正确性

1. 可执行文件 `dlc` 用于检查你的代码是否使用了非法操作以及操作数是否符合要求。

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

2. 在 `datalab-handout` 文件夹下使用 `make` 命令，会编译出可执行文件 `btest`，用于检查你的代码是否能返回正确的结果。

```
make
./btest
```

也可以用以下命令一键测试并得到一个参考得分。

```
./driver.pl
```

需要注意的是，`driver.pl` 给出的分数并不是你在本实验的最终得分。

关于本实验的其它未尽事宜请阅读 `README` 文件。

关于Linux系统的使用问题请尽量自行查询搜索引擎。对于实在没有办法自己解决的问题欢迎与助教交流。

值得注意

- `dlc` 和 `btest` 功能强大，使用 `-h` 查看其参数选项。
- 如果 `btest` 输出的你的答案和你代码应该得到的答案不符，有可能是编译器优化问题，尝试使用 `volatile` 关键字。
- `make` 还会同时编译 `fshow` 和 `ishow` 两个可执行文件，学会使用这两个程序可以加快你调试错误函数的速度。
- 所有和浮点数相关的题目均允许使用 `if`, `do`, `while`, `for`, `switch` 等条件控制和循环语句。

提交文件

- 你可以将代码提交到 `ics.men.ci` 并实时得到评测反馈。你的最终提交以你在网页上所有提交中**得分最高**的为准。
- 除源代码之外，你还需要撰写一份实验报告来解释你每个函数的实现思路，并提交至OBE。
- **实验报告必须为pdf格式，文件名只能使用ASCII字符。**

注意：在OBE上你只需要提交实验报告，不需要也不应该提交其他东西。

评分标准

你在本次实验的得分将由你**代码的实现情况**和**实验报告的情况**共同决定。

`ics.men.ci` 的排行榜将设置 `===95===`、`===90===` 和 `===Prof===` 三位虚拟选手，代码部分具体评分标准如下：

每题使用的运算符数量	本题得分
少于 <code>===95===</code>	100
等于 <code>===95===</code>	95
少于等于 <code>===90===</code>	90
少于等于 <code>===Prof===</code>	85
少于等于 <code>1.5*===Prof===</code>	60
少于等于 <code>2*===Prof===</code>	30
大于 <code>2*===Prof===</code>	0

代码部分的总分为每题得分的加权和，也就是排行榜上的分数。

注意：请所有同学确保自己的 `ics.men.ci`、`ics.ruc.rvalue.moe`、`obe.ruc.edu.cn` 这三个网站的密码均已设为强密码（至少8位，大小写字母、数字、符号均出现）。去年有人被偷作业，今年如果再发现直接按故意泄露处理。