

Lab1 Datalab 问题列表

约定:

位于低字节部分的第一个字节为第 0 个字节，位于最低有效位的位为第 0 位。由约定可知，一个 32 位的整数从低到高一共有 4 个字节，分别为第 0, 1, 2, 3 个字节，一个 32 位整数一共有 32 个位，分别为第 0, 1, 2, ..., 31 位。如 0x12345678 的第 0 个字节为 0x78。

小技巧:

- 多利用 btest 的测试用例进行修改;
- 可以随意定义变量来减少操作符个数;
- $a2 = a << 1$, $a4 = a << 2$ 以此类推;

nuaa_question1 tmin.c (10 分)

```
tmin - return minimum two's complement integer
```

使用位运算获取对 2 补码的最小 int 值

nuaa_question2 isTmax.c (10 分)

```
isTmax - returns 1 if x is the maximum, two's complement number, and 0 otherwise
```

通过位运算计算是否是补码最大值,是就返回 1, 否则返回 0

- 例如: `isTmax(0x7FFFFFFF) = 1`
- 例如: `isTmax(0) = 0`

nuaa_question3 allOddBits.c (20 分)

```
allOddBits - return 1 if all odd-numbered bits in word set to 1
```

判断所有奇数位是否都为 1, 是就返回 1, 否则返回 0

- 例如: `allOddBits(0xFFFFFFFF) = 0`
- 例如: `allOddBits(0xAAAAAAAA) = 1`

nuaa_question4 conditional.c (20 分)

```
conditional - same as x ? y : z
```

使用位级运算实现 C 语言中的 `x?y:z` 三目运算符

- 例如: `conditional(2,4,5) = 4`

nuaa_question5 float_twice.c (10 分)

float_twice - Return bit-level equivalent of expression $2*f$ for floating point argument f . Both the argument and result are passed as unsigned int's, but they are to be interpreted as the bit-level representation of single-precision floating point values.

- * When argument is NaN, return argument

给予一个无符号整数表示的浮点数 `uf` (你可以认为 `uf` 具有浮点数的比特级结构), 返回浮点参数 `uf` 的表达式 $2*uf$ 的位级等效值

nuaa_question6 float_f2i.c (30 分)

float_f2i - Return bit-level equivalent of expression $(int) f$ for floating point argument f . Argument is passed as unsigned int, but it is to be interpreted as the bit-level representation of a single-precision floating point value.

- * Anything out of range (including NaN and infinity) should return `0x80000000u`.

给予一个无符号整数表示的浮点数 `uf` (你可以认为 `uf` 具有浮点数的比特级结构), 函数返回它对应的强制类型转换后的整数 (具有整数的比特级结构), 即实现表达式 `(int) uf`