

Дата лаб: Битүүдийг манипуляцилах

ICSI251 / 2025-2026 намрын улирал

1 Удиртгал

Энэхүү даалгаврын гол зорилго нь бүхэл тоо (integer) болон хөвөгч таслалтай тоо (floating-point number)-ны битийн түвшний дотоод бүтэц, илэрхийлэлтэй илүү гүнзгий танилцахад оршино. Битийн түвшинд тоонууд хэрхэн хадгалагдаж, боловсруулагддагийг ойлгох нь програмчлалын суурь мэдлэгийг бататгах чухал алхам юм. Дасгалууд нь эхэндээ хийсвэр мэт санагдаж болох ч тэдгээрийг гүйцэтгэх явцад та битийн талаар илүү нарийн ойлголттой болж, логик сэтгэлгээ болон програмчлалын ур чадвараа хөгжүүлэх болно.

2 Даалгаврын удирдамж

Даалгаврыг бие даан гүйцэтгэнэ. Даалгаврыг цахимаар хураан авна.

3 Заавар

Файл татах: Энд `datalab-handout.tar` файлыг хэрхэн тараахыг тайлбарласан догол мөрийг оруулсан.

Эхлээд `datalab-handout.tar` файлыг Linux үйлдлийн системтэй хамгаалагдсан(protected) хавтсанд хуулна. Дараа нь дараах тушаалыг (command) өгнө:

```
unix> tar xvf datalab-handout.tar
```

Энэ нь хавтас доторх файлуудыг задлах тушаал юм. Эдгээр файлуудаас зөвхөн `bits.c` файлд даалгаврыг гүйцэтгэнэ.

`bits.c` файлд 13 программчлалын бодлого байх ба тус бүрийн эх загвар (skeleton) бэлэн байгаа. Таны даалгавар бол эдгээр функцийн загварыг integer төрлийн бодлогуудын хувьд зөвхөн шулуун кодоор (straight-line code — давталт, нөхцөлт

оператор ашиглахгүйгээр) болон зөвшөөрөгдсөн цөөн тооны C хэлний арифметик болон логикийн оператор ашиглан гүйцээх явдал юм.

Тодруулбал, танд зөвхөн дараах найман операторыг ашиглахыг зөвшөөрнө:

```
! ~ & ^ | + << >>
```

Цөөн хэдэн бодлогод энэ жагсаалтыг хязгаарладаг. Мөн 8 битээс урт тогтмолыг ашиглахыг хориглоно. Нарийвчилсан дүрмүүд болон хүссэн кодчиллын хэв маягийн талаарх хэлэлцүүлгийг bits.c дээрх тайлбараас харна уу.

4 Бодлогууд

Хүснэгт 1 нь битийн багцууд дээр ажиллах, тэдгээрийг өөрчлөх болон шалгах үйлдлүүдийг гүйцэтгэх функцүүдийн цогцыг тайлбарлаж буй. “Үнэлгээ” багана нь тухайн бодлогын хүндрэлийн түвшинг оноогоор илэрхийлдэг бол “Оп хязгаар” багана нь тухайн функцийг хэрэгжүүлэхдээ ашиглах боломжтой операторын дээд хязгаарыг зааж өгдөг. Функц бүрийн шаардлагатай үйл ажиллагаа болон нарийн дүрмийг тодруулахын тулд bits.c файл дахь тайлбаруудыг ашиглах хэрэгтэй. Мөн түүнчлэн tests.c файлд буй тестийн функцууд нь таны бичих ёстой функцуудын зөв ажиллах зарчмыг жишээгээр харуулсан лавлагаа болно. Гэвч эдгээр тестийн функцууд нь тухайн бодлогын кодчиллын дүрмүүдийг баримталдаггүйг анхаарах шаардлагатай.

Нэр	Тайлбар	Үнэлгээ	Макс оп
bitXor(x,y)	$x \parallel y$ зөвхөн & ба ~ ашиглан.	1	14
tmin()	Хамгийн бага two's complement бүхэл тоо	1	4
isTmax(x)	Үнэн, зөвхөн x хамгийн том two's comp. бүхэл тоо бол.	1	10
allOddBits(x)	Үнэн, зөвхөн x-ийн бүх сондгой дугаартай битүүд 1 бол.	2	12
negate(x)	-x буцаана - оператор ашиглахгүйгээр.	2	5
isAsciiDigit(x)	Үнэн хэрэв $0x30 \leq x \leq 0x39$.	3	15
conditional	$x ? y : z$ адил	3	16
isLessOrEqual(x, y)	Үнэн хэрэв $x \leq y$, бусад тохиолдолд худал	3	24
logicalNeg(x)	!x тооцоолно ! оператор ашиглахгүйгээр.	4	12
howManyBits(x)	x-ийг two's comp-д илэрхийлэх хамгийн бага битийн тоо.	4	90
floatScale2(uf)	f.p. аргумент f-ийн хувьд $2*f$ -ийн битийн түвшний эквивалент буцаана.	4	30
floatFloat2Int(uf)	f.p. аргумент f-ийн хувьд (int)f-ийн битийн түвшний эквивалент буцаана.	4	30
floatPower2(x)	Бүхэл тоо x-ийн хувьд 2.0^x -ийн битийн түвшний эквивалент буцаана.	4	30

Хүснэгт 1: Datalab бодлогууд. Хөвөгч таслалттай бодлогуудын хувьд, утга f нь unsigned бүхэл тоо uf-тэй ижил битийн төлөөлөл бүхий хөвөгч таслалттай тоо юм.

Хөвөгч таслалттай бодлогуудын хувьд, та зарим түгээмэл ганц нарийвчлалтай хөвөгч таслалттай үйлдлүүдийг хэрэгжүүлнэ. Эдгээр бодлогуудын хувьд, стандарт удирдлагын бүтэц (нөхцөлт, давталт) ашиглахыг зөвшөөрнө, мөн int болон unsigned өгөгдлийн төрлүүдийг, дурын unsigned болон бүхэл тоон тогтмолуудыг ашиглаж болно. Union, struct, эсвэл массив ашиглаж болохгүй. Хамгийн чухал нь, хөвөгч таслалттай өгөгдлийн төрөл, үйлдэл, тогтмол ашиглаж болохгүй. Үүний

оронд, хөвөгч таслалттай оролт нь функц руу unsigned төрөлтэйгээр дамжуулагдаж, буцаах хөвөгч таслалттай утга нь unsigned төрөлтэй байна. Таны код нь заагдсан хөвөгч таслалттай үйлдлүүдийг хэрэгжүүлэх битийн манипуляцийг гүйцэтгэх ёстой. Хавсаргасан fshow програм нь хөвөгч таслалттай тооны бүтцийг ойлгоход тусална. fshow-г компайл хийхийн тулд, handout хавтас руу шилжиж дараах командыг бичнэ:

```
unix> make
```

Та fshow-г ашиглан дурын битийн хэв маяг хөвөгч таслалттай тоо болж юу илэрхийлэхийг харж болно:

```
unix> ./fshow 2080374784
```

```
Floating point value 2.658455992e+36
```

```
Bit Representation 0x7c000000, sign = 0, exponent = f8, fraction = 000000
```

```
Normalized. 1.0000000000 X 2(121)
```

Мөн fshow-д hexadecimal болон хөвөгч таслалттай утгуудыг өгч, тэдгээрийн битийн бүтцийг задлан үзэж болно.

5 Үнэлгээ

Оноо хамгийн ихдээ 67 оноо тооцогдох бөгөөд дараах байдлаар хуваарилагдана:

- 36 оноо — Зөв байдал (Correctness points)
- 26 оноо — Гүйцэтгэл (Performance points)
- 5 оноо — Загвар, бичгийн хэв (Style points)

Зөв байдал: Шийдэх ёстой 13 бодлого тус бүрт 1–4 оноо хүртэл хүндрэлийн үнэлгээ оноосон бөгөөд нийт жинлэгдсэн нийлбэр нь 41 оноо болно. Таны функцуудыг btest програм ашиглан үнэлнэ. Функц бүх тестийг амжилттай давбал тухайн бодлого бүрэн оноо авна, бусад тохиолдолд оноо авахгүй.

Гүйцэтгэл: Хичээлийн энэ үе шатанд бидний гол анхаарал нь зөв хариу авахад байгаа боловч, кодоо аль болох товч, энгийн байлгах шаардлагатай. Зарим бодлогыг муйхар аргаар (brute force) шийдэж болно, гэхдээ илүү ухаалаг шийдэл гаргахыг хүсэж байна. Тиймээс функц бүрт ашиглаж болох операторын дээд хязгаарыг тогтоосон. Энэ хязгаар нь маш уян хатан бөгөөд зөвхөн үр дүн муутай, хэтэрхий удаан шийдлийг илрүүлэхэд зориулагдсан. Хязгаарласан операторын тоог баримталсан зөв функц бүрт 2 оноо өгөх болно.

Загвар, бичгийн хэв: Сүүлийн 5 оноог таны шийдлийн бичгийн хэв, тайлбарын үндсэн дээр өгөх болно. Код ойлгомжтой байх. Коммент нь мэдээлэл өгөхүйц, гэхдээ заавал их байх шаардлагагүй.

Ажлаа авто-үнэлгээ хийх

Бид autograding хэрэгслүүдийг handout фолдерт оруулсан — үүнд btest, dlc, болон driver.pl багтсан — таны ажлын зөв байдлыг шалгахад зориулагдсан.

- **btest**: Энэ програм нь bits.c дахь функцуудын зөв байдлыг шалгана. Компайл болон ашиглах команд:

```
unix> make
unix> ./btest
```

bits.c файлыг өөрчилсөн тохиолдолд btest-г дахин компайл хийх хэрэгтэй. Функцуудыг нэг бүрчлэн шалгахыг зөвлөж байна. Зөвхөн нэг функц шалгахын тулд -f флаг ашиглана:

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

Тодорхой аргументуудыг -1, -2, -3 флагуудыг ашиглан өгч болно. Жишээ нь:

```
unix> ./btest -f bitAnd -1 7 -2 0xf
```

Илүү дэлгэрэнгүй заавартай танилцахын тулд README файлыг үзнэ үү.

- **dlc**: MIT CILK бүлгийн ANSI C компайлерийн засварласан хувилбар бөгөөд кодын дүрэмд нийцэж байгаа эсэхийг шалгана. Энгийн ашиглалт:

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

Програм нь алдаа илрүүлээгүй бол дуугүй ажиллана. Алдаа нь:

- Хориотой оператор ашигласан
- Хэт олон оператор ашигласан
- Integer puzzle-д non-straightline код байгаа гэх мэт

Функц бүрт ашигласан операторын тоог гаргахын тулд -e флагийг ашиглана:

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

Бусад тушаалын сонголтуудыг үзэхийн тулд:

```
./dlc -help
```

- **driver.pl**: Энэ нь btest болон dlc-ийг ашиглан таны шийдлийн зөв байдал ба гүйцэтгэлийн оноог автоматаар тооцдог драйвер програм. Аргумент шаардахгүй:

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

Багш driver.pl-ийг ашиглан таны ажлыг үнэлнэ.

6 Илгээх заавар

Өөрийн ‘bits.c’ файлыг Teams-ийн даалгавар өгсөн хэсэгт илгээнэ.

7 Зөвлөгөө

- <stdio.h> толгой файлыг (header) bits.c-д оруулах хэрэггүй. Учир нь энэ нь dlc-г будилуулж, ойлгомжгүй алдааны мэдэгдэл гаргадаг. Гэхдээ debugging-ийн зорилгоор printf-ийг bits.c-д ашиглаж болно, <stdio.h> оруулаагүй ч болох ба gcc анхааруулга(warning) хэвлэх боловч тоохгүй байж болно.
- dlc програм нь C++ эсвэл gcc-ээс илүү хатуу C дүрэм мөрдүүлдэг. Тухайлбал, бүх зарлалууд (declarations) нь блок дотор () байрлаж, блокын эхэнд, ямар нэг зарлал биш statement-ээс өмнө байх ёстой. Жишээ нь дараах код дээр алдаа гарна:

```
1     int foo(int x)
2     {
3         int a = x;
4         a *= 3;      /* Statement that is not a declaration */
5         int b = a;   /* ERROR: Declaration not allowed here */
6     }
```

4 дүгээр мөрд зарлал биш байна. Харин 5 дугаар мөрөнд зарлал хийж болохгүй. Алдаа гарна.