Kvalifikatori znaka služe za izbor načina predstave brojeva, u smislu njihove označenosti. U slučaju da se koristi *unsigned* kvalifikator kao prefiks primitivnog tipa, takav tip se može koristiti isključivo za predstavu pozitivnih (neoznačenih) brojeva. Sa druge, strane, kada se koristi *signed* kvalifikator, moguća je predstava i pozitivnih i negativnih (označenih) brojeva. Ukoliko kvalifikator znaka izostavi, podrazumeva se upotreba *signed* kvalifikatora, odnosno označenih brojeva. Takođe, kvalifikatori znaka se koriste isključivo kod celobrojnih tipova (*int* ili *char*), dok je njihova primena kod realnih tipova zabranjena. Primer upotrebe ovih kvalifikatora je dat u nastavku.

```
unsigned int var1 = 10000;
signed int var2 = 10;
```

Listing 17.13: Način upotrebe kvalifikatora znaka

Rad sa ove dve vrste kvalifikatora je značajno olakšan u slučaju korišćenja biblioteke za rad sa celobrojnim tipovima – *stdint.h*, budući da su oni već inkorporirani u tipove definisane u okviru ove biblioteke. Njeno objašnjenje je dato u nastavku.

Integer tipovi podataka

Osnovni celobrojni tipovi podataka kao što su *char*, *int*, *long* i ostali se u opštem slučaju prevode od strane kompajlera na različite načine u zavisnosti od pratforme za koju se kompajliraju. Primer ove razlike jeste to da se tip int prevodi u 32-bitni (4-bajtni) označeni celobrojni tip za 32-bitne ili 64-bitne procesore, dok se u slučaju 8-bitnog ATmega328P procesora koji sadrži Arduino Uno ovaj tip prevodi u 16-bitni (2-bajtni) označeni celobrojni tip podataka. Kako će u opštem slučaju broj bajtova koji zauzima neki tip podataka zavisiti od platforme, upotreba osnovnih tipova podataka može dovesti do grešaka u prekoračenju opsega prilikom kompajliranja programa za različite platforme. Kako bi se ovaj problem izbegao, prilikom razvoja programa se upotrebljava standardna biblioteka `<stdint.h>`.

Biblioteka `<stdint.h>` sadrži definicije standardnih označenih i neoznačenih tipova podataka čije su dužine fiksirane, kao i najveće vrednosti koje one podržavaju. Sintaksa definisanih tipova za definisanje celobrojne tipove je `intN_t`, dok za neoznačene tipove ona ima oblik `uintN_t`, gde N predstavlja broj biti koji reprezentuje ovaj tip. Broj N može imati vrednosti 8, 16, 32 ili 64. U zavisnosti od broja bita i toga da li je tip označen ili ne, definišu se sledeći tipovi podataka:

- Označeni tipovi:
  - `int8_t`
  - `int16_t`
  - `int32_t`
  - `int64_t`
- Neoznačeni tipovi:

- `uint8_t`
- `uint16_t`
- `uint32_t`
- `uint64_t`

Za svaki od navedenih tipova makroima su definisane granica opsega vrednosti, tako se za označeni tip definiše minimum i maksimum, dok se za neoznačene tipove definiše samo maksimalna vrednost (podrazumeva se da je minimalna vrednost 0). Makroi koji definišu ove vrednosti su navedeni unutar biblioteke i imaju sledeći oblik:

```
/* Minimum of signed integral types.  */
#define INT8_MIN    (-128)
#define INT16_MIN   (-32767-1)
#define INT32_MIN   (-2147483647-1)
#define INT64_MIN   (-__INT64_C(9223372036854775807)-1)

/* Maximum of signed integral types.  */
#define INT8_MAX    (127)
#define INT16_MAX   (32767)
#define INT32_MAX   (2147483647)
#define INT64_MAX   (__INT64_C(9223372036854775807))

/* Maximum of unsigned integral types.  */
#define UINT8_MAX   (255)
#define UINT16_MAX  (65535)
#define UINT32_MAX  (4294967295U)
#define UINT64_MAX  (__UINT64_C(18446744073709551615))
```

Pored tipova podataka koji imaju tačno definisane fiksne vrednosti, bez obzira na odabranu platformu za koju se kompajliraju, definišu se i tipovi podataka koji će imati minimalnu širinu koja je navedena. Ovi tipovi su definisani kako za označene tako i za neoznačene vrednosti, gde je sintaksa `int_leastN_t` i `uint_leastN_t` respektivno. Podaci definisani pomoću ovih tipova će imati minimalnu širinu N bita, gde mogu imati više u zavisnosti od platforme za koju se kompajliraju. Ukoliko, na primer, platforma podržava samo `uint32_t` i `uint64_t` tipove, tada će prilikom definicije podatka upotrebom `uint8_t` tipa nastati greška. Kako bi se ova greška izbegla, često se upotrebljava celobrojni tip minimalne šririne, u ovom slučaju `uint_least8_t`.

Za tipove podataka minimalne širine, definisane su pomoću makroa najuže granice opsega koje podržavaju. Imena ovih makroa su `INT_LEASTN_MAX` za gornju granicu opsega označenog celobrojnog tipa minimalne širine N bita, zatim `INT_LEASTN_MIN` za donju granicu istog tipa i konačno `UINT_LEASTN_MAX` kao gornja granica neoznačenog tipa širine N bita. Kompletan pregled i opis svih tipova unutar ove biblioteke dostupan je na sledećoj stranici:

`https://pubs.opengroup.org/onlinepubs/009696799/basedefs/stdint.h.html`