

Nokkur æfingaverkefni sem ekki á að skila.

1. Gerðu eftirfarandi verkefni sem eru í námsefninu á netacad.com:

- Lab 4.4.1 (1) String manipulation: pangrams
- Lab 2.4.23 (1) Counting bits (the ones)
- Lab 2.4.23 (2) Bitwise palindromes

2. Útfærðu eftirfarandi föll og notaðu **bitaðgerðir** (e. *bitwise operator*) við lausn þeirra:

1. `iHastafi` sem tekur inn textastreng og skilar til baka streng þar sem búið er að setja alla bókstafi í inntaksstrengnum í hástafi. Athuga að gera viðeigandi ráðstafanir varðandi tölustafi, bil og annað sem ekki telst til bókstafa. Dæmi um notkun:

```
string texti = "abc 123 Def";
cout << iHastafi(texti); // ætti að skrifa út ABC 123 DEF
```

2. `iLagstafi` sem virkar eins og fallið hér fyrir ofan nema skilar streng með öllum bókstöfum í lágstöfum. Dæmi um notkun:

```
string texti = "ABC 123 Def";
cout << iLagstafi(texti); // ætti að skrifa út abc 123 def
```

3. `vixlaStafasetur` sem breytir hástöfum í lágstafi og lágstöfum í hástafi. Dæmi um notkun:

```
string texti = "Abc 123 dEf";
cout << vixlaStafasetur(texti); // ætti að skrifa út aBC 123 Def
```

3. Skrifðu forrit sem getur reiknað út og birt undirnet (e. *subnet*). Forritið á að biðja um upphafsnet (til einföldunar er það slegið inn án punkta), fjölda netbita og í hversu mörg net á að skipta netinu upp í. Forritið reiknar svo út nýju netin og sýnir allar helstu upplýsingar um hvert net, þ.e. netnúmerið (e. *network address*), fyrstu og síðustu nothæfu IP töluna, víðvarps númerið (e. *broadcast address*) og undirnetsmátið (e. *subnet mask*). Forritið skrifar þessar upplýsingar svo snyrtilega á skjáinn. Dæmi um virkni:

Sláðu inn upphafsnet: **192 168 1 0**

Fjöldi netbita í upphafsneti: **24**

Fjöldi neta sem á skipta upphafsnetinu í: **4**

IP net	Fyrsta nothæfa	Síðasta nothæfa	Broadcast	Maski
192.168.1.0	192.168.1.1	192.168.1.62	192.168.1.63	255.255.255.192
192.168.1.64	192.168.1.65	192.168.1.126	192.168.1.127	255.255.255.192
192.168.1.128	192.168.1.129	192.168.1.190	192.168.1.191	255.255.255.192
192.168.1.192	192.168.1.193	192.168.1.254	192.168.1.255	255.255.255.192

Byrjaðu á að útfæra eftirfarandi fjögur föll:

- `int ip2int(int* ipFylki)` Fallið tekur inn bendi á fjögurra staka heiltölufylki og skilar 32 bita heiltölu (má vera unsigned) sem er samsett úr tölunum í fylkinu. Þetta er gert þannig að fyrsta stakið í fylkinu er skrifað í 8 fyrstu bitana í heiltölunni, annað stakið í næstu 8 bita o.s.frv. Dæmi, fallið tekur inn fylki sem inniheldur stökin 192 (**11000000**₂), 168 (**10101000**₂), 1 (**00000001**₂) og 0 (**00000000**₂), það skilar þá 3232235776 (**11000000101010000000000100000000**₂) unsigned (eða -1062731520 signed).
- `void ip2fylki(int* ipFylki, int ipTala)` Fallið tekur við bendi á heiltölufylki og svo IP tölu sem er geymd í heiltölubreytu. Fallið tekur svo fyrstu átta bitana úr `ipTala` breytunni og setur í stak núll, næstu átta bitana í stak eitt o.s.frv. Fallið gerir sem sagt akkúrat andstöðuna við það sem að fallið `ip2int` gerir.
- `void synaIpTolu(string texti, int* ipFylki)` Fallið tekur á móti texta og fjögurra staka fylki sem inniheldur eina IP tölu. Fallið skrifar svo út textann og svo IP töluna með punktum á milli octetanna.

- `void synaIpTolu(string texti, int ipTala)` Fallið er fjölbundin (*e. overloaded*) útgáfa af fallinu hér að ofan en tekur við IP tölunni í heiltöluformi.

Til að reikna út netin er þægilegast að nota *magic-number* aðferðina. Hún gengur út á að finna gildi síðasta bitans í nýja subnet maskanum sem er 1. Áður en það er gert þarf að finna út hversu marga bita þarf fyrir nýju netin. Það má gera með eftirfarandi:

Fjöldi nýrra netbita = $\lceil \log_2 x \rceil$, þar sem x er fjöldi neta. Dæmi, ef t.d. á að skipta neti upp í 4 undirnet þá má reikna: $\lceil \log_2 4 \rceil = 2$, það þarf þá 2 netbita fyrir fjögur net.

Til að finna sætið sem síðasti 1 bitinn er í má nota:

Síðasta sætið = $32 - (\text{netbitar í upphafsneti} + \text{fjöldi nýrra netbita})$. Dæmi: Upphafsnet 24 bitar, fjöldi bita fyrir ný net 2. Síðasti bitinn er þá $32 - (24 + 2) = 6$

Að lokum er síðasta sætið svo notað til að reikna út *magic-number* með því að setja töluna 2 í veldið sem kom út úr útreikningum hér að ofan:

magic-number = $2^{\text{Síðasta sætið}}$. Dæmi, síðasta sætið er 6 þá er *magic-number* = 2^6 , eða 64.

Nú er hægt að finna hvert net með því að hækka alltaf um *magic-number*-ið. Fyrir fyrstu nothæfu IP-töluna bætist svo einn við, fyrir seinustu nothæfu bætist (*magic-number* - 2) við og fyrir víðvarpsnúmerið bætist (*magic-number* - 1) við.

Til að finna undirnetsmátið (*e. subnet mask*) má taka 32 bita heiltölu og setja alla 32 bitana á einn og hliðra svo til vinstri (*e. left shift*) um $32 - (\text{netbitar í upphafsneti} + \text{fjöldi nýrra netbita})$ sæti.

Föllin `ceil`, `log2` og `pow` eru í `cmath` safninu.