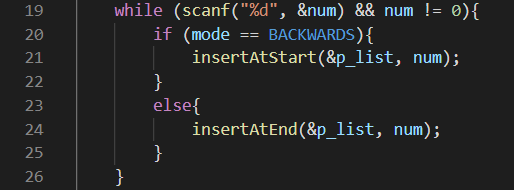
**Kodavimo** **standartas**

Darbo autorius – 4 grupės studentas Armintas Pakenis. Darbas atliktas C kalba. Sukūrimo data 2020-12-07, paskutinė pakeitimo data: 2020-12-15.

Formatas

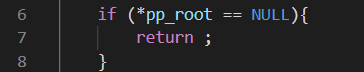
1. Kodas turi būti rašomas taip, kad ji būtų galima kompiliuoti C89 versija.
2. Eilutės ilgis neturėtų būti ilgesnis nei 90 simbolių.
3. Riestiniai skliaustais visada turi būti apskliaustas kodo blokas, einantis po *while, do, for, if, else, switch* sakinių; net jei po *while, do, for, if, else, switch* sakinių seka tik vienas arba tuščias sakinys, jį reikia vis tiek apskliausti.
   1. Kairinis riestinis skliaustas ‘{’ turi atsirasti *while, do, for, if, else, switch* sakinio pabaigoje, o nenaujoje eilutėje. Dešininisriestinis skliaustas ‘}’ turi atsirasti naujoje eilutėje, vienodai sulygiuotas su *while, do, for, if, else, switch* žodžiu.

Tinkamas apskliaudimas:



1. Vengti vartoti *goto*, jei galima išsiversti būtų be jo.
2. Skaitomumui pagerinti gali būti įterptos tuščios eilutės tarp bet kurių prasminių eilučių, tačiau negalima įterpti daugiau nei 3 iš eilės einančių tuščių eilučių.
3. Lyginant kintamuosius su konstantomis, kintamasis turėtų būti kairėje, jei taip įmanoma užrašyti.

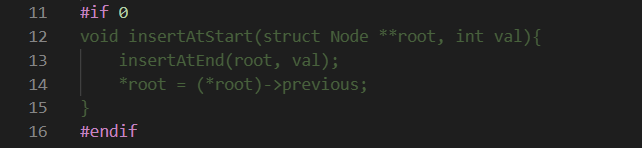
Tinkamai parašytas lyginimas:



Komentarai

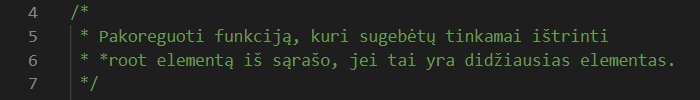
1. Komentarai gali būti ir vienos eilutės ilgio, ir ilgesni. Komentaras turi būti tarp ‘/\*‘ ir ‘\*/‘.
2. Komentare negali būti pavartota trigrafų ar kitokių preprocesoriaus token‘ų.
3. Nenaudojamas kodas negali būti užkomentuojamas naudojant ‘/\*‘ ir ‘\*/‘. Tą reikėtų daryt panaudojant #if 0 <...kodas...> #endif.

Tinkamai užkomentuotas kodas:



1. Kelių eilučių komentarai turi būti pradedami naujoje, t. y. tuščioje eilutėje tik simboliu ‘/\*’. Tolimesnės eilutės turi prasidėti simboliu ‘**\***’, vienodai išlygiuotu su pirmosios eilutės ‘\*’; po žvaigždutės turi sekti tarpas. Komentaro paskutinėje eilutėje turi būti tik ‘\*/’.

Pavyzdys:

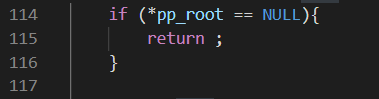


1. Komentarai turi būti prasmingi, vadovautis principu *DRY* (Don‘t Repeat Yourself), t. y., kodas neturi būti pakartojimas komentare.

Tarpai

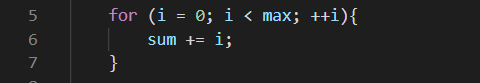
1. Po raktinių žodžių while, for, switch, if ir return turi sekti tarpas.

Tinkamai suformatuota eilutė, kur panaudotas minėtas iš raktinių žodžių:



1. **Prieš** ir **po** operatorių =, \*=, /=, +=, -=, &=, %=, |=, ^=, ~=, != , *ternary* operatorių ?, :, ir binarių operatorių +, -, \*, /, %, <, <=, >, >=, ==,!=, <<, >>, &, |, ^, &&, || turi būti vienas tarpas.
2. Po unarinių operatorių +, -, ++, --, ! , ~ neturi sekti joks tarpas. Po struktūrų operatorių -> ir . neturi sekti joks tarpas.
3. Rodyklių operatoriai \* ir & turi būti parašyti su tarpu iš kairės (pvz.:‘ \*‘) deklaracijose (aprašuose). Kitais atvejais tarpo po šių rodyklių operatorių dėti tarpo nereikia.
4. Po masyvui skirtų operatorių ( [ ir ] ) neturi eiti joks tarpas prieš ar po, nebent to reikalauja kita tarpams skirta taisyklė.
5. Po skliausto ‘(‘ neturi iškart eiti joks tarpas; prieš skliaustą ‘)‘ neturi iškart būti jokio tarpo.
6. Po kiekvieno kablelio ‘ , ‘ turi eiti tarpas.
7. Po kiekvieno kabliataškio ‘;‘ turi eiti tarpas, jei jis skiria sakinių elementus (pvz. for cikle). Kitu atveju, jei kabliataškis terminuoja sakinį, tarpo po kabliataškio dėti nereikia.

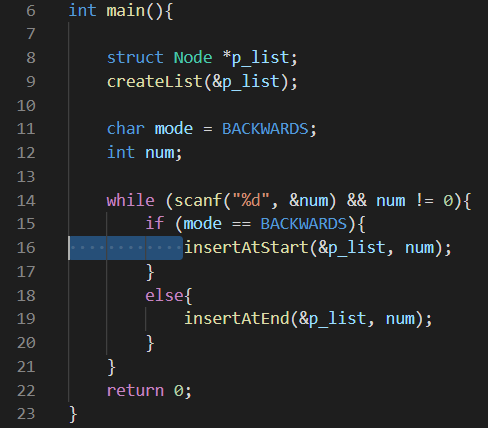
Tinkamai padėti tarpai po kabliataškių:



Atitraukimas

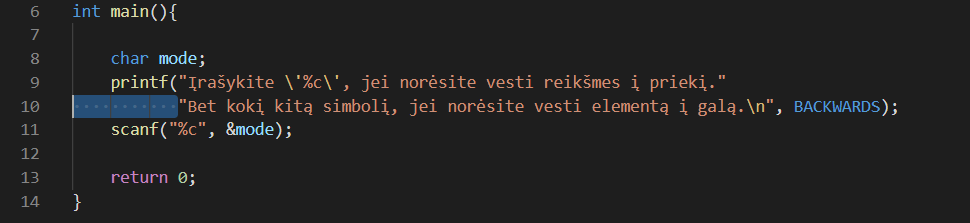
1. Blokų lygiai nustato atitraukimo lygį. Bloko lygis nustatomas pagal tai, kelių kitų blokų viduje jis yra. Pirmame bloke eilutės turėtų prasidėti 4 tarpais nutolę nuo pradžios. Jei tame bloke egzistuoja kitas blokas, tai tame kitame bloke eilutės turėtų prasidėti tarpais nutolę nuo pradžios, kur n – bloko lygis.

Tinkamai atitrauktos kodo eilutės:



1. Jei atsiranda eilutė, kuri neįvykdo formato 2 punkto (t. y. ilgesnė už 90 simbolių), antrą jos dalį galima rašyti kitoje eilutėje minimaliai atitraukus pagal viršiau pateiktą punktą. Papildomi tarpai gali būti pridėti, jei tai gerina kodo skaitomumą.

Pavyzdys:



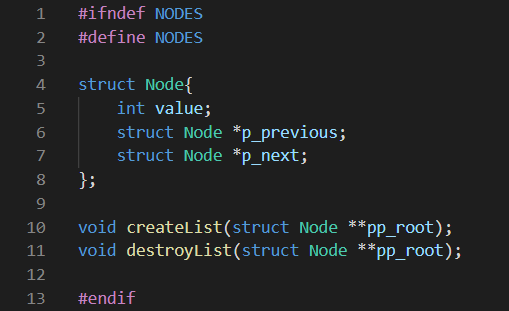
Paaiškinimas: 10 eilutėje minimaliai reikėjo atitraukti 4 tarpais (vadovaujantis 1 punktu). Tačiau antras punktas leidžia pridėti papildomų tarpų, jei tai gerina skaitomumą ir komanda netilpo vienoje eilutėje, nes viršijo 90 simbolių.

1. Atitraukiama turi būti tik tarpais (simboliais, kurių ASCII kodas yra tik ir tik 32).

Moduliai

1. Modulių pavadinimai:
   1. Modulių pavadinimai gali būti sudaryti tik iš mažųjų lotyniškų raidžių, skaičių ir ‘\_‘ (angl. underscore) simbolio.
   2. Po modulių pavadinimo turi sekti ‘.h‘, jei tai antraštės (angl. header) failas; ‘.c‘, jei tai pagrindinis (angl. source) failas.
   3. Modulių failai negali vadintis taip pat kaip C standartinės bibliotekos failai.
   4. Modulis, kuris turi savo viduje funkciją *int main()* privalo būti pavadintas ‘main.c‘.
2. Antraštės (angl. header) failai:
   1. Kiekvienam *source* failui, kuris nėra ‘main.c‘, privalo būti po vieną ir tik vieną antraštės failą, kuris turėtų tokį patį pavadinimą.
   2. Kiekvienas antraštės failas turėtų turėti pasikartojimo apsaugą panaudojant preprocesoriaus komandas. Po #ifndef ir #define turi sekti antraštės pavadinimas didžiosiomis raidėmis su pabaigoje pridėtais ‘\_H‘ simboliais, kaip pateikta žemiau pavyzdyje.

Pavyzdys:



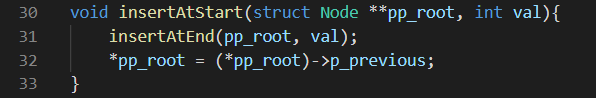
* 1. Į antraštės failus gali būti įtrauktos tik funkcijų, konstantų, išvestinių duomenų tipų deklaracijas, bei panaudos preprocesoriaus komandos, kaip #include, įtraukiančios standartines bibliotekas.

1. *Source* failai:
   1. Kiekvienas ne ‘main.c‘ source failas privalo įtraukti (#include) antraštės failą su tokiu pačiu pavadinimu.
   2. *Source* failo aprašų eiliškumas turi būti toks, tačiau ne kiekvienas punktas privalo būti *source* faile: bendras komentaras apie programą ar/ir fialą, #include komandos, #define komandos, funkcijų apibrėžimai (angl. definitions).

Pavadinimai

1. Nei vienas kintamojo, parametro, funkcijos ar išvestinio tipo pavadinamas negali turėti tokio vardo, kuris būtų C kalbos raktinis žodis. Pavadinimai privalo būti kuo labiau prasmingi.
2. Visi pavadinimai privalo būti susieti su anglų kalba.
3. Pavadinimai negali būti ilgesni nei 32 simboliai ir negali būti trumpesni nei 3 simboliai.
4. Funkcijos pavadinamas negali prasidėti *underscore* simboliu, taip pat negali prasidėti didžiąja raide.
5. Norint atskirti atskirus prasminius žodžius funkcijoje, tai reikia padaryti sekančiojo žodžio pirmąją raidę paverčiant didžiąja.

Pavyzdys:



1. Kintamųjų vardai gali būti sudaryti tik iš mažųjų lotyniškų raidžių, skaičių ir *underscore* simbolio. Didžiųjų raidžių būti negali.
2. Kintamieji privalo prasidėti bent kuria nors lotyniškos abėcėlės raide, o žodžių junginiai turi būti sujungiami *underscore* simboliu. Pvz.: int last\_session, passport\_id.
3. Jei kintamasis yra rodyklė į kažką, jo vardas turi prasidėti ‘p\_‘. Jei kintamasis yra rodyklė į rodyklė, pavadinimas privalo prasidėti ‘pp\_‘ ir t. t.
4. Patartina sukurti kintamąjį tik tada, kai jo reikės.
5. ‘,‘ operatorius neturėtų būti naudojamas deklaracijose. Tos pačios kintamuosius reikėtų deklaruoti iš naujos eilutės.
6. Konstantų, apibrėžtų per #define, pavadinimai privalo būti tik iš didžiųjų raidžių ir jei tai nėra antraštės failas, negali pabaigoje turėti ‘\_H‘.
7. Išvestinių duomenų tipų pavadinimai privalo prasidėti didžiąja raide ir negali turėti jokių skaitmenų.