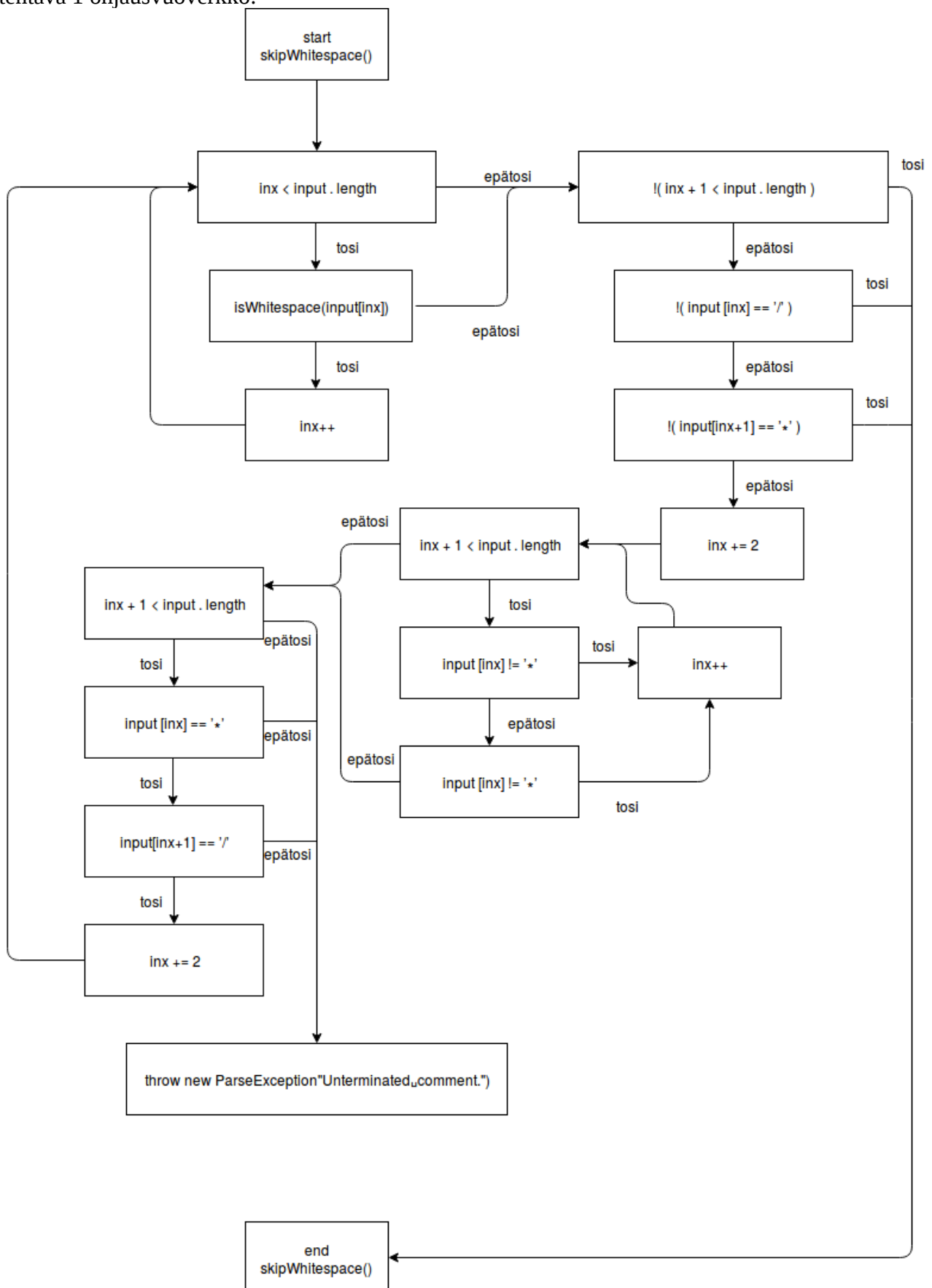


## Harjoitus 1 – Henri Haverinen

tehtävä 1 ohjausvuoverkko:



tehtävä 1 peruslohkokattavuuden kriteerin täyttävä yksikkötestauksen testijoukko:

1 tapaus: `inx = 0`, `input = " /*kissa*koira*/kilpikonna"`

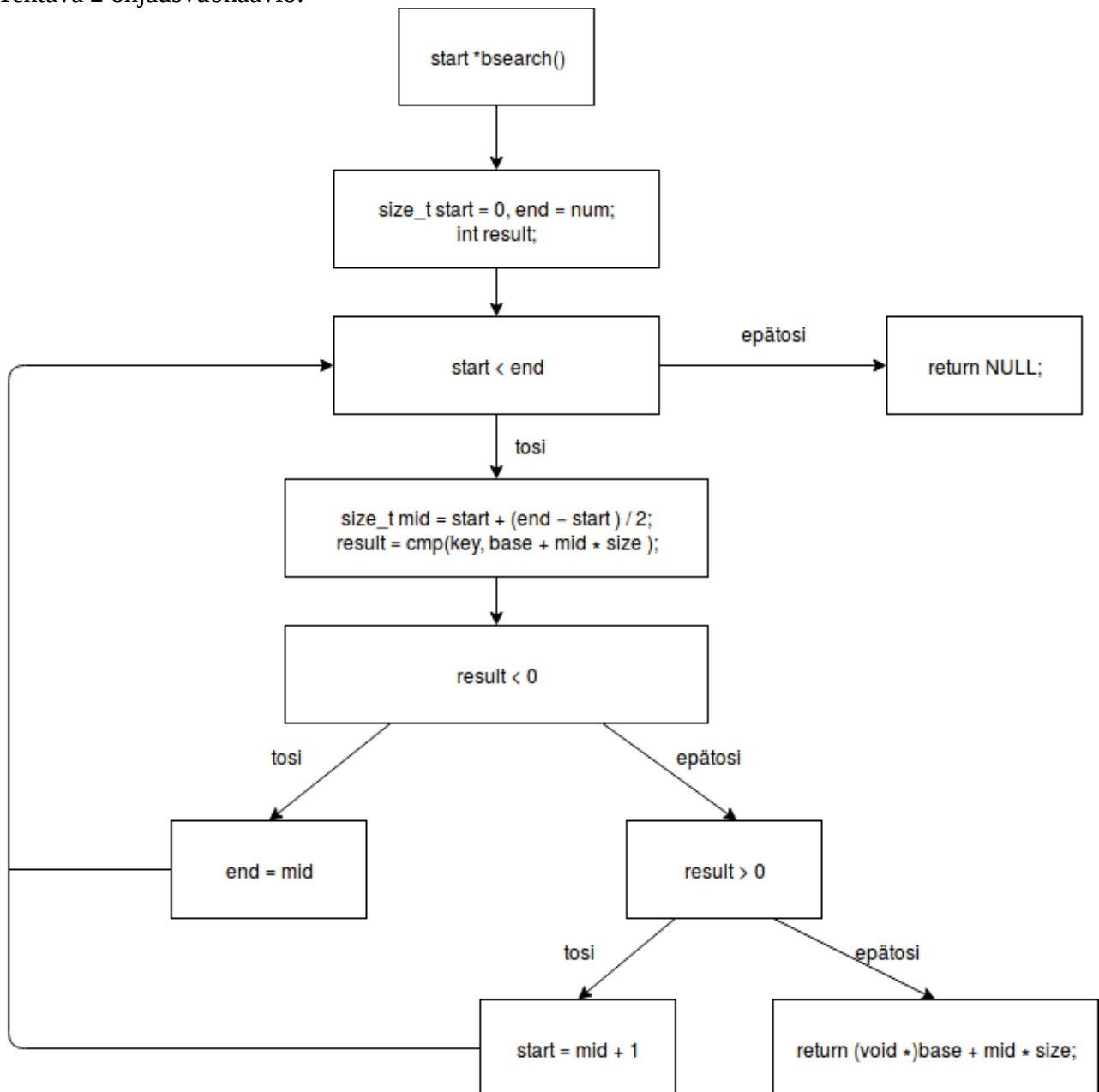
- kattaa ensimmäisen 2. tason `while`-silmukan kokonaan
- kattaa ensimmäisen `if`-lausekkeen ehdot ensimmäisellä kierroksella
- kattaa toisen 2. tason `while`-silmukan kokonaan
- kattaa `if-else` -lausekkeen `if`-osuuden kokonaan
- toisella kierroksella ensimmäinen `if`-lauseke johtaa "`end skipWhitespace()`" -loppusolmuun

2. tapaus: `inx = 0`, `input "kissa/*`

- johtaa tilanteeseen, jossa `if-else` -lausekkeesta toteutuu `else` osuus, jolloin suoritus päättyy "`throw new ParseException("Unterminated_ comment.")`" -loppusolmuun.

Edellä esitettyjä testitapauksia vastaavat testipolut kattavat yhdessä jokaisen ohjausvuoverkon solmun, ja täten täyttävät peruslohkokattavuuden kriteerin.

Tehtävä 2 ohjausvuokaavio:



tehtävä 2 peruslohkokattavuuden kriteerin täyttävä yksikkötestauksen testijoukko:

1. tapaus:

- key = "b"
- base = "abc"
- num = 3
- size = sizeof (char)
- cmp = strcmp

1. tapauksen yhdistelmän pitäisi kattaa if-"else-if"-else" -rakenteen else-if ja else -osuudet. Ohjelman suoritus päättyy "return (void \*)base + mid \* size;" -loppusolmuun.

2. tapaus:

- key = "d"
- base = "abc"
- num = 3
- size = sizeof (char)
- cmp = strcmp

2. tapauksen yhdistelmä päättyy ensimmäisen if-lausekkeen sisään, ja edelleen while-silmukan ehdon ulkopuolelle päättyen "return NULL;" -loppusolmuun.

Edellä esitettyjä testitapauksia vastaavat testipolut kattavat yhdessä jokaisen ohjausvuoverkon solmun, ja täten täyttävät peruslohkokattavuuden kriteerin.

NB. 2. tehtävässä olleen bsearch() -funktion ymmärtämiseksi hyödynnetty seuraavaa nettisivua: [https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_bsearch.htm](https://www.tutorialspoint.com/c_standard_library/c_function_bsearch.htm)