

[Dashboard](#) / [My courses](#) / [ITB IF2110 1 2324](#) / [Praktikum 12 \(17 November - 24 November\)](#) / [Pohon Biner - Praktikum](#)

<b>Started on</b>	Friday, 24 November 2023, 1:01 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 24 November 2023, 2:27 PM
<b>Time taken</b>	1 hour 25 mins
<b>Grade</b>	<b>40.00</b> out of 100.00

Question **1**

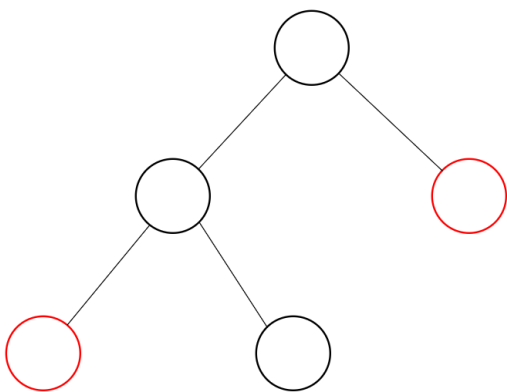
Correct

Mark 40.00 out of 40.00

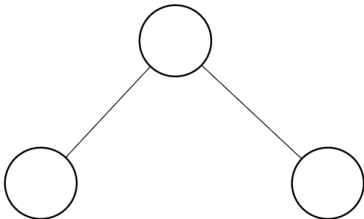
Time limit	1 s
Memory limit	64 MB

Burbir memiliki sebuah pohon biner. Burbir hanya menyukai pohon biner yang setiap nodenya tidak memiliki child sama sekali atau memiliki child kanan dan kiri. Dengan kata node tidak boleh hanya memiliki child left atau hanya memiliki child right. Oleh karena itu, Burbir ingin mengubah pohon yang dimilikinya agar dapat memenuhi kriteria yang disukainya dengan menambahkan nol atau lebih node. Tentukan jumlah node minimum yang perlu ditambahkan Burbir.

Contohnya yaitu misalkan diberikan pohon biner berikut, dengan node berwarna hitam merupakan pohon biner masukan dan node berwarna merah merupakan node yang perlu ditambahkan Burbir. Pada kasus ini jumlah minimum node yang perlu ditambahkan adalah 2.



Sedangkan pada kasus berikut ini tidak ada node yang perlu ditambahkan Burbir, sehingga jumlah minimum node yang perlu ditambahkan adalah 0.



Tambahkan fungsi di bawah ini ke dalam header [bintree.h](#) dari praprak kalian, lalu implementasikan fungsi di bawah ini dalam file bintree.c lalu kumpulkan hitung.zip yang berisi bintree.c dan [bintree.h](#)

```
int hitungNode(BinTree root);  
// Menerima sebuah pohon biner  
// Mengembalikan jumlah minimum node yang perlu ditambahkan
```

C

 [hitung.zip](#)

Score: 40

Blackbox

Score: 40

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	4	Accepted	0.00 sec, 1.63 MB

No	Score	Verdict	Description
2	4	Accepted	0.00 sec, 1.60 MB
3	4	Accepted	0.00 sec, 1.60 MB
4	4	Accepted	0.00 sec, 1.62 MB
5	4	Accepted	0.00 sec, 1.64 MB
6	4	Accepted	0.00 sec, 1.64 MB
7	4	Accepted	0.00 sec, 1.66 MB
8	4	Accepted	0.00 sec, 1.49 MB
9	4	Accepted	0.00 sec, 1.64 MB
10	4	Accepted	0.00 sec, 1.64 MB

Question **2**

Not answered

Marked out of 30.00

Time limit	1 s
Memory limit	64 MB

Tuan Hak memiliki banyak bebek. Bebek-bebek Tuan Hak ditempatkan di kandang-kandang pada lokasi yang berbeda-beda. Represetasi lokasi kandang dapat digambarkan sebagai node, dengan edge adalah antar node menggambarkan bahwa dua kandang saling bertetangga. Untuk menempuh kandang yang berada di child, Tuan Hak perlu melintasi kandang yang berada di parent terlebih dahulu baru pergi ke kandang di child.

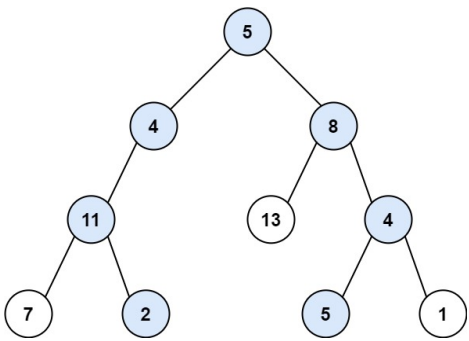
Tuan Man yang membuka warung bebek Ari Bolme membutuhkan bebek sebanyak  $n$  setiap harinya. Tuan Man meminta Tuan Hak untuk mengirimkan kepadanya  $n$  bebek. Tuan Hak ingin mengetahui, rute dari kandang-kandang yang **harus dia lalui hingga kandang yang paling akhir** (tidak memiliki child lagi) sedemikian sehingga dia berhasil mendapatkan bebek **tepat sebanyak  $n$** . Keluarkan hasil perkalian antara jumlah bebek dalam kandang untuk rute yang harus dia lalui. Jika terdapat lebih dari satu rute, keluarkan hasil penjumlahan antara hasil perkalian antara jumlah bebek dalam kandang untuk rute yang harus dia lalui (lihat contoh untuk lebih jelasnya). Hasil akhir dimodulo  $10^4$ .

Tambahkan fungsi di bawah ini ke dalam header [bintree.h](#) dari praprak kalian, lalu implementasikan fungsi di bawah ini dalam file `bintree.c` lalu kumpulkan **bebek.zip** yang berisi `bintree.c` dan [bintree.h](#)

```
int hitungBebek(BinTree root, int n);  
  
// Menerima sebuah pohon biner  
  
// Mengembalikan penjumlahan dari hasil kali antara bebek-bebek pada rute
```

Contoh:

Misalkan diberikan sebuah pohon biner seperti pada gambar dibawah dan Tuan Man membutuhkan bebek sebanyak 22 buah.



Terdapat dua rute yang mungkin, yaitu  $[5,4,11,2]$  dan  $[5,8,4,5]$ . Keluarkan  $((5 * 4 * 11 * 2) + (5 * 8 * 4 * 5)) \% (10^4) = 1240$

Hint:

$(a * b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$

$(a + b) \bmod n = (a \bmod n) + (b \bmod n)$

C

Question **3**

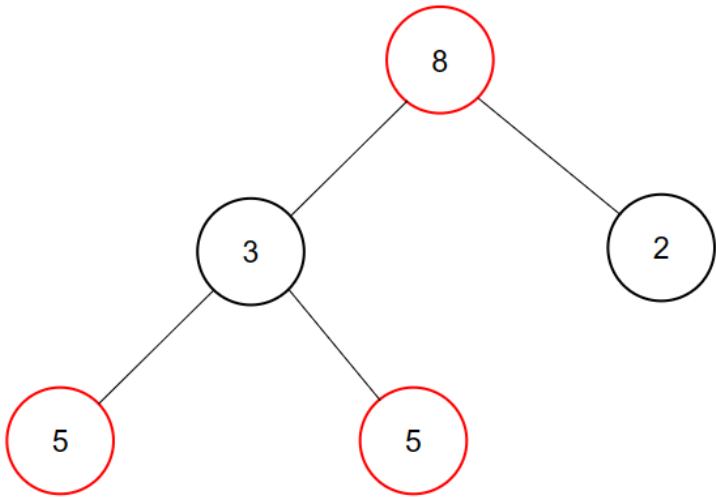
Incorrect

Mark 0.00 out of 30.00

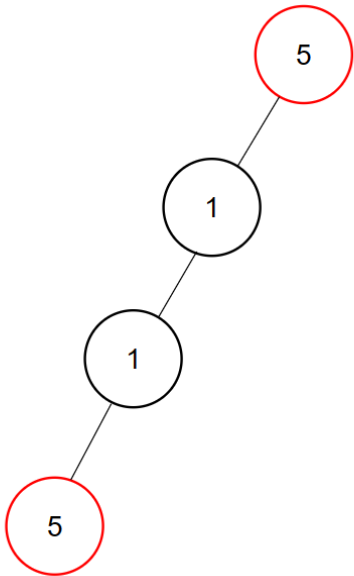
Time limit	1 s
Memory limit	64 MB

Burbir adalah seorang pencuri. Ia ingin mencuri pada sebuah perumahan. Perumahan bisa direpresentasikan dalam bentuk pohon biner, dengan node menggambarkan rumah dan edge antar rumah menggambarkan bahwa node saling bertetangga. Untuk mencegah dirinya tertangkap, jika ia mencuri pada sebuah rumah maka ia tidak bisa mencuri rumah yang bertetangga. Jika INFO dari node menyatakan jumlah uang yang bisa dicuri pada sebuah rumah, tentukan jumlah maksimal uang yang bisa dicuri dari perumahan tersebut.

Contohnya yaitu misalkan diberikan pohon biner tersebut. Node yang berwarna merah menyatakan rumah yang akan dicuri Burbir, sehingga total maksimum uang hasil curiannya adalah  $8 + 5 + 5 = 18$



Untuk pohon biner berikut ini total maksimum uang hasil curiannya adalah  $5 + 5 = 10$ .



Tambahkan fungsi di bawah ini ke dalam header [bintree.h](#) dari praprak kalian, lalu implementasikan fungsi di bawah ini dalam file bintree.c lalu kumpulkan **jumlah.zip** yang berisi bintree.c dan [bintree.h](#)

```
int hitungUang(BinTree root);  
// Menerima sebuah pohon biner yang menyatakan denah perumahan  
// Mengembalikan jumlah maksimum uang yang bisa dicuri Burbir
```

C

[jumlah.zip](#)

Score: 0

Blackbox

Score: 0

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	0	Wrong answer	0.00 sec, 1.55 MB
2	0	Wrong answer	0.00 sec, 1.63 MB
3	0	Wrong answer	0.00 sec, 1.71 MB
4	0	Wrong answer	0.00 sec, 1.64 MB
5	0	Wrong answer	0.00 sec, 1.65 MB
6	0	Wrong answer	0.00 sec, 1.50 MB
7	0	Wrong answer	0.00 sec, 1.66 MB
8	0	Wrong answer	0.00 sec, 1.71 MB
9	0	Wrong answer	0.00 sec, 1.63 MB
10	0	Wrong answer	0.00 sec, 1.64 MB

[◀ Pohon Biner - Pra Praktikum](#)

Jump to...

[Pohon Biner - Praktikum \(extended\) ▶](#)