

[Dashboard](#) / [My courses](#) / [ITB_IF2110_1_2526](#) / [Praktikum 10](#) / [Post Praktikum 10](#)

Started on	Friday, 28 November 2025, 7:16 PM
State	Finished
Completed on	Friday, 28 November 2025, 8:24 PM
Time taken	1 hour 8 mins
Marks	400.00/400.00
Grade	10.00 out of 10.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: identical.c

Lengkapi fungsi pada file [identical.c](#)

Spesifikasi Fungsi:

```
#include <stdio.h>
#include <stdlib.h>
#include "bintree.h"

boolean isIdentical(BinTree P, BinTree Q);
```

Deskripsi:

Fungsi isIdentical harus menentukan apakah dua binary tree identik atau tidak. Dua binary tree dikatakan identik apabila memiliki struktur yang sama dan nilai pada node-node yang bersesuaian juga sama.

Contoh Aplikasi Fungsi:

Tree 1:

```
      (1)
     /  \
    (2)  (3)
```

Tree 2:

```
      (1)
     /  \
    (2)  (3)
```

Output: 1 (true)

Tree 1:

```
      (5)
     /  \
    (3)  (8)
   / \  \
  (1) (4) (9)
```

Tree 2:

```
      (5)
     /  \
    (3)  (8)
   / \  \
  (1) (2) (9)
```

Output: 0 (false)

- Catatan:**
- Jumlah node $n \in [1, 10^4]$
 - Nilai tiap node $\in [-10^5, 10^5]$

C

 [identical.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.72 MB
2	10	Accepted	0.00 sec, 1.65 MB
3	10	Accepted	0.00 sec, 1.51 MB
4	10	Accepted	0.00 sec, 1.61 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 2.41 MB
7	10	Accepted	0.00 sec, 2.05 MB
8	10	Accepted	0.00 sec, 1.98 MB
9	10	Accepted	0.00 sec, 1.80 MB
10	10	Accepted	0.00 sec, 1.99 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Dengan adt [bintree.h](#)
Lengkapi fungsi pada [problem.c](#) !

C

 [good-nodes.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.65 MB
2	10	Accepted	0.00 sec, 1.71 MB
3	10	Accepted	0.00 sec, 1.65 MB
4	10	Accepted	0.00 sec, 1.62 MB
5	10	Accepted	0.00 sec, 1.65 MB
6	10	Accepted	0.00 sec, 1.65 MB
7	10	Accepted	0.00 sec, 1.71 MB
8	10	Accepted	0.00 sec, 1.65 MB
9	10	Accepted	0.00 sec, 1.51 MB
10	10	Accepted	0.00 sec, 1.60 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: levelMaxSum.c

Header: #include "bintree.h"

Anda diminta mengimplementasikan fungsi **levelMaxSum** yang menerima akar (root) dari sebuah *binary tree* dan mengembalikan level dengan jumlah nilai node (*sum*) terbesar. Level akar adalah 1, level anak-anaknya 2, dan seterusnya. Jika terdapat lebih dari satu level dengan jumlah maksimum yang sama, fungsi harus mengembalikan level terkecil (paling atas). **Lengkapi saja file berikut fungsi pada [problem.c](#) !**

Spesifikasi Fungsi:

```
#include <stdio.h>
#include <stdlib.h>
#include "bintree.h"

int levelMaxSum(BinTree p);
```

Deskripsi:

Fungsi levelMaxSum akan melakukan penelusuran level demi level (menggunakan *Breadth-First Search*) untuk menghitung jumlah nilai pada setiap level pohon. Level dengan total nilai terbesar akan dikembalikan sebagai hasil.

Contoh Aplikasi Fungsi:

Pohon:

```
      (1) -> level 1
     /  \
    (7) (0) -> level 2 (MAX)
   /  \
  (7) (-8) -> level 3
```

Penjelasan:

Level 1: 1

Level 2: 7 + 0 = 7

Level 3: 7 + (-8) = -1

Output:

2

Pohon:

```
      (989) -> level 1
       \
      (10250) -> level 2 (MAX)
     /   \
  (98693) (-89388) -> level 3
       \
      (-32127) -> level 4
```

Penjelasan:

Level 1: 989

Level 2: 10250

Level 3: 98693 + (-89388) = 9305

Level 4: -32127

Output:

2

Catatan:

- Jumlah node $n \in [1, 10^4]$.
- Nilai node $\in [-10^5, 10^5]$.
- Jika ada lebih dari satu level dengan sum maksimal yang sama, kembalikan level terkecil (paling atas).

C

 [levelMaxSum.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.64 MB
2	10	Accepted	0.00 sec, 1.67 MB
3	10	Accepted	0.00 sec, 1.63 MB
4	10	Accepted	0.00 sec, 1.66 MB
5	10	Accepted	0.00 sec, 1.73 MB
6	10	Accepted	0.00 sec, 1.63 MB
7	10	Accepted	0.00 sec, 1.66 MB
8	10	Accepted	0.00 sec, 1.52 MB
9	10	Accepted	0.00 sec, 1.54 MB
10	10	Accepted	0.00 sec, 1.51 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: maxPathSum.c

Lengkapi fungsi pada file [maxPathSum.c!](#)

Spesifikasi Fungsi:

```
#include <stdio.h>
#include <stdlib.h>
#include "bintree.h"

int maxPathSum(BinTree root);
```

Deskripsi:

Fungsi maxPathSum harus menentukan nilai maksimum dari semua kemungkinan path pada binary tree. Path boleh:

- hanya terdiri dari satu node,
- berjalan ke kiri atau kanan saja,
- atau melalui tiga node: kiri → parent → kanan.

Contoh Aplikasi Fungsi:

Tree:

```
      (1)
     /  \
    (2)  (3)
```

Path terbaik:
2 + 1 + 3 = 6

Output: 6

Tree:

```
      (-10)
     /    \
    (9)    (20)
     /      \
   (15)     (7)
```

Path terbaik:
15 → 20 → 7 = 42

Output: 42

- Catatan:**
- Jumlah node $n \in [1, 10^4]$
 - Nilai tiap node $\in [-10^5, 10^5]$
 - Path tidak harus melalui root
 - Path tidak boleh melewati node yang sama lebih dari satu kali

C

 [maxPathSum.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	6	Accepted	0.00 sec, 1.65 MB
2	6	Accepted	0.00 sec, 1.52 MB
3	6	Accepted	0.00 sec, 1.71 MB
4	6	Accepted	0.00 sec, 1.67 MB
5	6	Accepted	0.00 sec, 1.66 MB
6	6	Accepted	0.00 sec, 1.67 MB
7	6	Accepted	0.00 sec, 1.58 MB
8	6	Accepted	0.00 sec, 1.63 MB
9	6	Accepted	0.00 sec, 1.56 MB
10	6	Accepted	0.00 sec, 1.65 MB
11	6	Accepted	0.00 sec, 1.61 MB
12	6	Accepted	0.00 sec, 1.65 MB
13	6	Accepted	0.00 sec, 1.61 MB
14	6	Accepted	0.00 sec, 1.63 MB
15	16	Accepted	0.00 sec, 1.64 MB

[◀ Praktikum 10](#)

Jump to...

[ADT Binary Tree ▶](#)