

# No.1-5

06 October 2020 8:02

## Question 1

Not yet answered

Marked out of 1.00

Flag question

Edit question

Perhatikan method rekursif berikut untuk menjawab pertanyaan 1-5.

```
1 void myRecFunction(int number) {  
2     if (number < 0) {  
3         System.out.println("-");  
4         myRecFunction(Math.abs(number));  
5     } else if (number < 10) {  
6         System.out.println(number);  
7     } else {  
8         myRecFunction(number / 10);  
9         System.out.println(number % 10);  
10    }  
11 }
```

Pada pemanggilan myRecFunction(19041), ada berapa baris luaran yang dicetak?  
Tuliskan langsung angkanya saja; contoh: 10

Answer:

## Question 2

Not yet answered

Marked out of 1.00

Flag question

Edit question

Nilai kondisi yang menunjukkan kondisi berhenti method tersebut adalah ...



Select one:

- ☐ Ketika number bernilai sama dengan 0
- ☐ Ketika number bernilai kurang dari 10
- ☐ Ketika number bernilai kurang dari 0
- ☐ Ketika number bernilai lebih dari 10

**Question 3**

Not yet answered

Marked out of 1.00

 Flag question Edit question

Jika nilai parameter number merupakan bilangan positif lebih dari 9, pemanggilan method `myRecFunction(int number)` terjadi pada baris ...



Select one:

- ☐ Baris 4 dan 8
- ☐ Hanya baris 8
- ☐ Hanya baris 4
- ☐ Hanya baris 6

**Question 4**

Not yet answered

Marked out of 1.00

 Flag question Edit question

Yang dilakukan oleh method tersebut adalah ...


Select one:

- ☐ Mencetak bilangan bulat dari number sampai 1 (menurun)
- ☐ Mencetak setiap digit pada bilangan number
- ☐ Mencetak semua bilangan kelipatan 10 yang ada pada range 1 sampai number (inklusif)
- ☐ Mencetak bilangan bulat dari 1 sampai number (menaik)

**Question 5**

Not yet answered

Marked out of 1.00

 **Flag question** **Edit question**

Jika N merepresentasikan nilai parameter number, kompleksitas waktu method rekursif tersebut adalah ...

Select one:

- ☐  $O(N)$
- ☐  $O(1)$
- ☐  $O(\log N)$
- ☐  $O(N \log N)$

## No.6


08 October 2020


5:34

### Question 6

Not yet answered

Marked out of 1.00

 Flag question

 Edit question

Perhatikan implementasi fungsi rekursif berikut.

```
int func(int n) {  
    if (n < 5) return n * 7;  
    else if (n < 10) return func(n-1) - 2;  
    else if (n < 50) return 2 * func(n/2);  
    else return func(n/3) + 5;  
}
```

Pada pemanggilan **func(100)**, berapa kali pemanggilan rekursif terjadi? (tidak termasuk pemanggilan func(100))

Tuliskan langsung angkanya saja; contoh: 10

Answer:

## No.7

08 October 2020 5:35

Perhatikan ketiga method rekursif berikut.

```
int recA(int x) {  
    if (x == 0) {  
        return x;  
    } else {  
        return x * recA(x - 1);  
    }  
}  
  
int recB(int y, int z) {  
    if (y == 0) {  
        return z;  
    } else {  
        return recB(y - 1, z - 1);  
    }  
}  
  
int recC(int n) {  
    if (n > 10) {  
        return n - 1;  
    } else {  
        return recC(recC(n + 5));  
    }  
}
```

*Handwritten notes:*  
- For recA: "tidak semua komputasi dilempar ke rec. case" and "bukan tail Rec."  
- For recB: "tidak diproses / dikalkulasi lg." and "tail Rec."  
- For recC: "tidak ada kalkulasi" and "tail Rec."

Pernyataan yang benar mengenai ketiga method tersebut adalah ...

Select one:

- ☐ recA, recB, dan recC bukan tail recursive
- ☐ recA dan recB merupakan tail recursive namun recC bukan tail recursive
- ☐ recA bukan tail recursive namun recB dan recC merupakan tail recursive
- ☐ recA, recB, dan recC merupakan tail recursive

Perhatikan ketiga method rekursif berikut.

```
int mystery(int m, int n) {
    if (n == 0) {
        return 1;
    } else if (n % 2 == 0) {
        return mystery(m, n / 2) * mystery(m, n / 2);
    } else {
        return m * mystery(m, n / 2) * mystery(m, n / 2);
    }
}
```

*m n input*  
*base*  
*conquer*  
*n genap*  
*n ganjil*  
*breakdown angkanya*

```
int tukarKoin(int koin[], int numKoin, int jumlahUang) {
    int[] table = new int[jumlahUang + 1];
    table[0] = 1;
    for (int i = 1; i < table.length; i++) {
        table[i] = 0;
        for (int j = 0; j < numKoin; j++) {
            for (int k = koin[j]; k <= jumlahUang; k++) {
                table[k] += table[k - koin[j]];
            }
        }
    }
    return table[jumlahUang];
}
```

*1, 5, 10*  
*uangnya*  
*tabel*  
*utk simpan info*  
*look up*  
*Informasi cara menyusun koin*  
*utk semua nilai dr 0 sd jumlahUang*

- Rde Greedy? ✓  
 - Backtracking? ✓  
 - Divide & Conq? ✓  
 - DP ✓

1 5 10

1 5 10

banyaknya cara menyusun koin

Uang	Cara
0	1
1	1
2	1
3	1
...	...
10	1

15  
 1 1 1 1 1 1 ...  
 2 5 1 1 1 1 1 ...  
 3 5 5 1 1 1 ...  
 4 ...  
 ...  
 7

```
void jobSelection(int startTimes[], int finishTimes[], int numActivities) {
    int i = 0;
    System.out.println("Do job from " + startTimes[i] + " to " + finishTimes[i]);
    numActivities--;
    int j = i++;
    while (j < startTimes.length && startTimes[j] < finishTimes[i]) {
        numActivities--;
        j++;
    }
    if (numActivities > 0) {
        jobSelection(Arrays.copyOfRange(startTimes, j, startTimes.length),
            Arrays.copyOfRange(finishTimes, j, finishTimes.length),
            numActivities);
    }
}
```

Pernyataan yang benar terkait ketiga method tersebut adalah ...

Select one:

- ☐ Method tukarKoin menerapkan paradigma greedy, sedangkan method jobSelection menerapkan paradigma backtracking
- ☒ Method mystery menerapkan paradigma divide and conquer, sedangkan method tukarKoin menerapkan paradigma dynamic programming
- ☐ Method mystery menerapkan paradigma dynamic programming, sedangkan method jobSelection menerapkan paradigma greedy
- ☐ Method mystery menerapkan paradigma dynamic programming, sedangkan method tukarKoin menerapkan paradigma greedy

Jika  $m$  menyatakan numKoin dan  $n$  menyatakan jumlahUang, kompleksitas waktu fungsi tukarkoin adalah ...


Select one:


- ☐  $O(\log n)$
- ☐  $O(n)$
- ☒  $O(mn)$
- ☐  $O(m \log n)$

#### Question 10

Not yet answered

Marked out of 1.00

 Flag question

 Edit question

Yang dilakukan oleh fungsi mystery adalah ...

Select one:

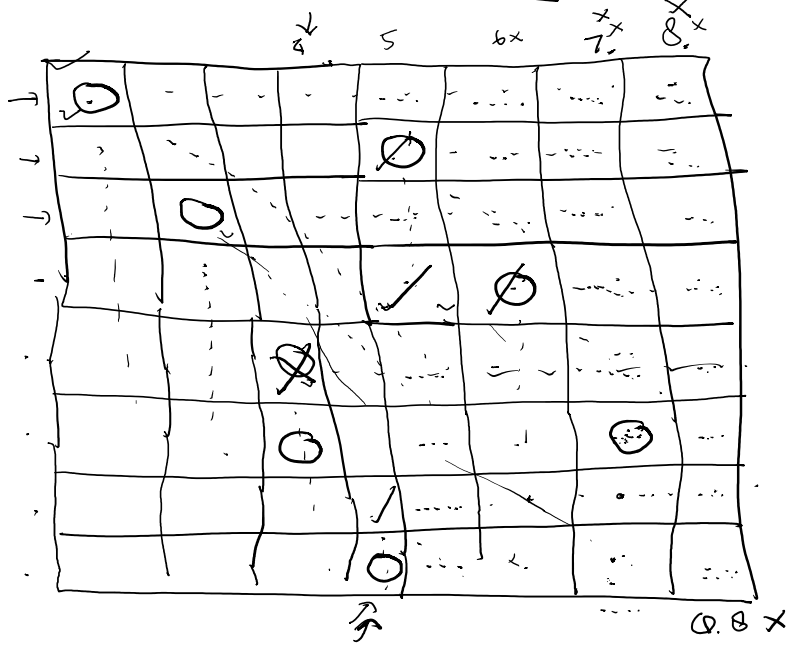
- ☐ Menentukan FPB dari bilangan  $m$  dan  $n$
- ☐ Menghitung nilai  $m$  dikalikan dengan  $n$
- ☒ Menghitung nilai  $m$  dipangkatkan dengan  $n$
- ☐ Mengubah bilangan  $m$  ke dalam basis  $n$

# N Queen

08 October 2020 8:16

8 Queen

Backtracking



X case =  $64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57$

8x8 (64) positi  
utk 8 queen

Queen		
1	1	1
2	2	3
3	3	4
4	4	5
5	5	
6	1	
7		
8		

x