

EASY

[001.SLL.prefor pencetakan string] Buatlah pseudocode untuk mencetak **Halo Dunia** sebanyak 3 kali tanpa menggunakan struktur perulangan.

Pencetakan harus dilakukan satu per satu.

Contoh solusi yang tidak diperkenankan:

```
output("Halo Dunia", "Halo Dunia", "Halo Dunia")
```

→ tidak diperkenankan karena mencetak sekaligus, tidak satu per satu

Program HaloDunia

Algoritma

... ..

[002.SLL.prefor pencetakan bilangan] Buatlah pseudocode untuk mencetak barisan bilangan dari 1 hingga 5 tanpa menggunakan struktur perulangan.

Pencetakan harus dilakukan satu per satu dengan hanya menggunakan satu variable, *i*, yang disediakan saja dan pencetakan hanya diperkenankan menggunakan instruksi ***output(i)*** saja.

Contoh solusi yang tidak diperkenankan:

output(i, i + 1, i + 2, i + 3, i + 4, i + 5)

→ tidak diperkenankan karena mencetak sekaligus, tidak satu per satu dan menggunakan instruksi pencetakan selain *output(i)*

Program Cetak

Kamus

... ..

Algoritma

i ← 1

output(i)

... ..

[003.SLL.prefor penjumlahan] Buatlah pseudocode untuk menjumlahkan bilangan dari 1 hingga 5 tanpa menggunakan struktur perulangan.

Penjumlahan harus dilakukan satu per satu dengan hanya menggunakan satu variable, *i*, yang disediakan saja.

Contoh solusi yang tidak diperkenankan:

$i \leftarrow 1 + 2 + 3 + 4 + 5$

→ tidak diperkenankan karena menjumlahkan sekaligus, tidak satu per satu

Program Penjumlahan
Kamus

.....

Algoritma

$i \leftarrow 1$

.....

[004.SLL.prefor penghitungan saldo akhir] Buatlah pseudocode untuk menghitung saldo akhir dari 3 buah transaksi tanpa menggunakan struktur perulangan. Tiap transaksi diminta dan dihitung satu per satu. Transaksi tanpa tanda (contoh: 50000) artinya pemasukan, transaksi dengan tanda minus (contoh: -15000) berarti pengeluaran. Cetak saldo akhir di akhir pseudocode.

Nilai tiap transaksi harus diminta dan dihitung satu per satu.

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
50000	60000
-15000	
25000	
120000	-30000
-100000	
-50000	

Program *HitungSaldoAkhir*
Kamus

... ..

Algoritma

... ..

[005.SLL.prefor perkalian] Buatlah pseudocode untuk mengalikan 4 buah bilangan bulat yang dimasukkan oleh *user* satu per satu. Perkalian dilakukan tanpa menggunakan struktur perulangan dan dihitung satu per satu. Cetak hasil perkalian keempat bilangan di akhir pseudocode.

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
5	360
3	
4	
6	
1	4
2	
1	
2	
1	

Program Perkalian

Kamus

... ..

Algoritma

... ..

EASY

[006.SLL.prefor perpangkatan] Buatlah pseudocode untuk menghitung hasil x^5 tanpa menggunakan *library math* maupun struktur perulangan. Nilai x didapatkan dari masukan *user*.

Perpangkatan harus dilakukan satu per satu.

Contoh solusi yang tidak diperkenankan:

$hasil_pangkat \leftarrow x * x * x * x * x$

→ tidak diperkenankan karena melakukan operasi perkalian sekaligus dalam satu instruksi, tidak satu per satu

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
2	32
10	100000

Program PangkatLima

Kamus

... ..

Algoritma

... ..

MEDIUM

[007.SLL.prefor Pencetakan Status Ganjil] Buatlah pseudocode untuk memeriksa apakah 3 bilangan bulat yang dimasukkan oleh *user* satu per satu adalah bilangan ganjil atau bukan. Kalau ganjil, algoritma akan mencetak **true**, bila bukan, **false**. Pencetakan status ganjil dilakukan tanpa menggunakan struktur perulangan.

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
5	true
13	true
2	false
11	true
9	true
9	true

Program StatusGanjil

Kamus

... ..

Algoritma

... ..

MEDIUM

[008.SLL.prefor Pencetakan Status Genap] Buatlah pseudocode untuk memeriksa apakah 3 bilangan bulat yang dimasukkan oleh *user* satu per satu adalah bilangan genap atau bukan. Kalau genap, algoritma akan mencetak **true**, bila bukan, **false**. Pencetakan status genap dilakukan tanpa menggunakan struktur perulangan.

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
5	false
13	false
2	true
4	true
2	true
12	true

Program StatusGenap

Kamus

... ..

Algoritma

... ..

MEDIUM

[009.SLL.prefor Pencetakan Status Faktor] Buatlah pseudocode untuk memeriksa apakah 3 pasang bilangan bulat yang dimasukkan oleh *user* satu per satu adalah pasangan faktor atau bukan. Kalau faktor, algoritma akan mencetak **true**, bila bukan, **false**. Pencetakan status faktor dilakukan tanpa menggunakan struktur perulangan.

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
4 8	true
9 3	true
6 7	false
1 3	true
3 5	false
5 2	false

Program StatusFaktor

Kamus

... ..

Algoritma

... ..

HARD

[010.SLL.prefor Status Terurut] Buatlah pseudocode untuk memeriksa apakah 10 buah bilangan bulat yang dimasukkan oleh *user* satu per satu terurut membesar (*ascending*) atau tidak. Bila terurut membesar, algoritma akan mencetak **true**, bila bukan, **false**. Pencetakan status terurut dilakukan tanpa menggunakan struktur perulangan. Gunakan variabel tidak lebih dari 5 buah!

Contoh *input/output*:

<i>Input</i>	<i>Output</i>
2	true
6	
7	
8	
9	
10	
13	
14	
17	
20	
1	false
2	
3	
4	
5	
6	
7	
8	
9	
8	

Program StatusTerurut

Kamus

... ..

Algoritma

... ..