

**KUMPULAN SOAL-SOAL
ALGORITMA DAN PEMROGRAMAN (4 sks)
Dr. Aris Puji Widodo, MT.**

Perhatikan:

- a. Semua Algoritma yang telah diselesaikan, kemudian diterjemahkan ke dalam bahasa pemrograman dengan nama program/sub program (termasuk file .c nya) harus sesuai dengan nama yang telah dispesifikasikan pada setiap soal.
- b. Setiap mahasiswa wajib membuat cek list progress yang dikerjakan dengan menggunakan template seperti dibawah ini.

Cek List Progress Kuliah Algoritma dan Pemrograman

Nama : _____

Nim : _____

No	Nama Program	Algoritma	Bahasa C	Keterangan
1.	Diisikan nama program	Diisikan [Ok Not]	Diisikan [Ok Not]	Diisikan Tgl Selesai Pengerjaan Algoritma: dd-mm-yyyy Bahasa C: dd-mm-yyyy Keterangan: {berikan kendala yang anda dapatkan (jika ada)}

Soal-soal

1. (**nama program: jarakGLBB**) Dibaca tiga buah nilai v_0 (kecepatan awal dalam meter per detik), t (waktu dalam detik), dan a (percepatan dalam meter per detik kuadrat) yang mewakili koefisien gerak lurus berubah beraturan. Harus dihitung dan dituliskan hasilnya, jarak (S dalam meter) yang ditempuh benda yang mengalami gerak lurus berubah beraturan dengan kecepatan awal v_0 dalam waktu t , dan percepatan a . Diketahui bahwa jarak adalah $S = v_0 * t + \frac{1}{2} * (a * t^2)$.

2. **(nama program: jarakPBola)** Dibaca dua buah nilai v_0 (kecepatan awal dalam meter per detik), dan t (waktu dalam detik), dengan g sebuah konstanta gravitasi (dalam meter per detik kuadrat) yang mewakili koefisien gerak parabola. Harus dihitung dan dituliskan hasilnya, jarak (y dalam meter) yang ditempuh benda yang mengalami gerak parabola dengan kecepatan awal v_0 dalam waktu t dengan gravitasi g . Diketahui bahwa jarak adalah $y = v_0 * t - \frac{1}{2} * (g * t^2)$.
3. **(nama program: gayaSentr)** Dibaca tiga buah nilai m (massa dalam kg), v (kecepatan dalam meter per detik), dan r (jari-jari dalam meter) yang mewakili koefisien gaya sentripetal. Harus dihitung dan dituliskan hasilnya, Gaya (F dalam Newton) yang terjadi dari massa m , kecepatan v , dan jari-jari r . Diketahui bahwa Gaya adalah $F = m * (v^2 / r)$.
4. **(nama program: volBolaKerct)** Dibaca satu buah nilai r (jari-jari lingkaran dalam meter). Harus dihitung dan dituliskan hasilnya, volume bola (V_b dalam meter kubik), dan volume kerucut (V_k dalam meter kubik) dari sebuah lingkaran dengan jari-jari r dan konstanta Φ sebesar 3.1415. Diketahui bahwa volume bola adalah $V_b = \frac{4}{3} * (\Phi * r^3)$ dan volume kerucut $V_k = \frac{1}{2} * \text{volume bola}$.
5. **(nama program: luasKellLayang)** Dibaca empat buah nilai s_1 (sisi satu layang-layang dalam meter), s_2 (sisi dua layang-layang dalam meter), d_1 (diagonal satu layang-layang dalam meter), dan d_2 (diagonal dua layang-layang dalam meter). Harus dihitung dan dituliskan hasilnya, luas layang-layang (Luas dalam meter persegi), dan keliling layang-layang (Kell dalam meter) dari sebuah layang-layang dengan sisi s_1 dan s_2 , serta diagonal d_1 dan d_2 . Diketahui bahwa luas layang-layang adalah $\text{Luas} = \frac{1}{2} * d_1 * d_2$, dan keliling layang-layang adalah $\text{Kell} = 2 * (s_1 + s_2)$.
6. **(nama program: CekProsesBil)** Buatlah teks algoritma jika diberikan sebuah bilangan integer sembarang N , Bilangan N tersebut dilakukan klasifikasi proses dengan melakukan output nilai N yang terakhir setelah diklasifikasi. Adapun proses klasifikasinya adalah sebagai berikut :
 - a. Jika N bilangan genap maka N ditambah dengan 3, Kemudian jika N ditambah 3 merupakan kelipatan 5 maka ditambahkan lagi dengan 5, tetapi jika tidak kelipatan 5 maka ditambahkan 2.
 - b. Jika N bilangan ganjil maka N ditambah dengan 2, Kemudian jika N ditambah 2 merupakan kelipatan 3 maka ditambahkan lagi dengan 4, tetapi jika tidak kelipatan 3 maka ditambahkan 1.
7. **(nama program: CekBilSemb)** Buatlah teks algoritma, jika diberikan sebuah inputan i bilangan integer sembarang, maka tuliskanlah ke layar bahwa bilangan i tersebut sebagai bilangan bulat positif atau nol atau bulat negative. Tetapi jika yang diinputkan bukan sebuah bilangan maka akan mengeluarkan pesan **"Bukan termasuk sebuah bilangan"**
8. **(nama program: CekHari)** Buatlah teks algoritma yang digunakan untuk menuliskan ke layar nama-nama Hari dari nomor hari, yaitu 1 s.d. 7 yang dibaca dari masukan keyboard. Teks

algoritma melakukan validasi terlebih dahulu apakah masukan dari keyboard **sudah benar atau belum**, yaitu apakah **nomor Hari sudah tepat 1 s.d. 7**. Jika **tidak tepat**, maka dikeluarkan **pesan kesalahan yaitu “Masukan nomor hari tidak tepat”**.

9. (**nama program: CekBulan**) Buatlah teks algoritma yang digunakan untuk menuliskan ke layar nama-nama bulan dari nomor bulan, yaitu 1 s.d. 12 yang dibaca dari masukan keyboard. Teks algoritma melakukan validasi terlebih dahulu apakah masukan dari keyboard **sudah benar atau belum**, yaitu apakah **nomor bulan sudah tepat 1 s.d. 12**. Jika **tidak tepat**, maka dikeluarkan **pesan kesalahan yaitu “Masukan nomor bulan tidak tepat”**.
10. (**nama program: CekSeriPositif**) Buatlah teks algoritma yang menerima 3 buah tahanan yang dimasukan menggunakan keyboard dan menghasilkan total tahanan jika dirangkai seri. **Ketiga masukan tahanan tidak boleh bernilai negatif**. Jika **salah satu masukan tahanan bernilai negatif**, maka diberikan **pesan kesalahan “Masukan tahanan tidak boleh negatif”**.
11. (**nama program: CekSegitiga**) Buatlah teks algoritma, jika diberikan 3 buah inputan bilangan integer positif (>0) sembarang yang merepresentasikan sisi-sisi segitiga. Tuliskanlah ke layar dari ke-3 inputan tersebut termasuk **Segitiga Sama Sisi**, atau **Segitiga Sama Kaki**, atau **Segitiga Sembarang**. Jika salah satu atau lebih inputan yang dimasukan adalah **bilangan Nol atau Negatif** (≤ 0), maka dilayar akan dikeluarkan pesan **“Terdapat nilai yang bukan sisi segitiga”**
12. Buatlah teks algoritma dari sebuah kalkulator Super Sederhana Kalk-SS, yang menerima masukan 2 buah bilangan bulat sembarang melalui keyboard, misalnya iA dan iB , dan menuliskan beberapa hasil operasi aritmatika terhadap kedua bilangan tersebut berdasarkan pilihan operasi menggunakan character sebagai berikut:
 - Jika pilihan = **a** : menuliskan menuliskan hasil $iA + iB$
 - Jika pilihan = **b** : menuliskan hasil $iA - iB$
 - Jika pilihan = **c** : menuliskan hasil $iA * iB$
 - Jika pilihan = **d** : menuliskan hasil $iA/iB \rightarrow$ hasilnya real/float
 - Jika pilihan = **e** : menuliskan hasil $iA \text{ div } iB \rightarrow$ hasilnya integer
 - Jika pilihan = **f** : menuliskan hasil $iA \text{ mod } iB$
 - Jika dimasukkan pilihan **menu character lain**, maka **dituliskan “Bukan pilihan menu yang benar”**

Contoh Masukan dan Keluaran

Masukan			Keluaran
A	B	pilihan operasi	
3	2	a	$A + B = 5$
3	2	b	$A - B = 1$
3	2	c	$A * B = 6$
3	2	d	$A / B = 1.5$
3	2	e	$A \text{ div } B = 1$
3	2	f	$A \text{ mod } B = 1$
3	2	h	Bukan pilihan menu yang benar

13. (**nama program: BiayaParkir**) Buatlah sebuah teks algoritma untuk menentukan besarnya biaya parkir yang dihitung berdasarkan lamanya parkir (bilangan integer >0) yang dimasukkan melalui keyboard. Perhitungan biaya parkir berdasarkan aturan sebagai berikut: Biaya parkir 2 jam pertama Rp. 2000,- dan per jam berikutnya dihitung Rp. 500,-

Contoh:

Lama parkir 1 Jam Biayanya sebesar Rp. 2000,-

Lama parkir 2 Jam Biayanya sebesar Rp. 2000,-

Lama parkir 4 Jam Biayanya sebesar Rp. 3000,-

14. (**nama program: HargaDiskon**) Buatlah teks algoritma untuk menghitung besarnya harga setelah diberikan diskon, jika diberikan masukan melalui keyboard untuk besaran jenis [hanya terdiri dari 'A', 'B', dan 'C'], dan harga (integer > 200 dan < 10.000) . Aturan pemberian berdasarkan jenis , sehingga jenis A diskon 10%, jenis B diskon 15%, dan jenis C diskon 20%.

Contoh:

Jenis=A, Harga =Rp. 5000,- sehingga harga nya sebesar Rp. 4500,-

15. (**nama program: UpahKerja**) Buatlah teks algoritma, jika diberikan masukan melalui keyboard golongan dalam integer dari 1 s/d 4, dan integer banyaknya jam kerja per minggu > 0, maka hitunglah upah yang diterima dengan menggunakan aturan bahwa jika jam kerja perminggu kurang dari 40 jam maka menggunakan aturan seperti yang diberikan di tabel dibawah ini. Tetapi jika jam kerja per minggu lebih dari 40 jam, maka yang 40 jam dihitung dengan aturan yg diberikan di tabel, dan kelebihannya dihitung untuk setiap kelebihan 1 jam dihitung 1.5 kali dari upah yang diberikan di tabel.

Tabel Upah

Golongan 1 dengan upah per jam 1.000 rupiah

Golongan 2 dengan upah per jam 1.500 rupiah

Golongan 3 dengan upah per jam 2.000 rupiah

Golongan 4 dengan upah per jam 2.500 rupiah

16. (**nama program: TunjAnak**) Buatlah sebuah teks algoritma, jika diberikan sebuah bilangan integer jumlah anak ≥ 0 , dan integer gaji pokok >0 , maka hitunglah besarnya tunjangan anak yang diberikan, dan kemudian tampilkan besarnya tunjangan anak tersebut. Aturan pemberian tunjangan adalah setiap anak akan diberikan tunjangan sebesar 10% dari gaji pokok, dan pemberian tunjangan hanya diberikan kepada 3 anak maksimum.

Contoh:

Jumlah anak=0 dan Gaji pokok = 1000 maka tunjangan anak= $0 \times (10\% \times \text{Gaji pokok}) = 0$

Jumlah anak=2 dan Gaji pokok = 1000 maka tunjangan anak= $2 \times (10\% \times \text{Gaji pokok}) = 200$

Jumlah anak=3 dan Gaji pokok = 1000 maka tunjangan anak= $3 \times (10\% \times \text{Gaji pokok}) = 300$

Jumlah anak=5 dan Gaji pokok = 1000 maka tunjangan anak= $3 \times (10\% \times \text{Gaji pokok}) = 300$

Jumlah anak=10 dan Gaji pokok = 1000 maka tunjangan anak= $3 \times (10\% \times \text{Gaji pokok}) = 300$

17. (**nama program: TarifPLN**) Buatlah sebuah teks algoritma, jika diberikan sebuah bilangan integer 1 s/2 sebagai golongan tarif, dan pemakaian daya listrik (dalam kWh) sebagai integer >0 , maka hitunglah besarnya tarif listrik yang dikenakan, kemudian tampilkan besarnya tariff tersebut.

Tarif listrik yang yang diberlakukan adalah:

Golongan 1 tarif : Rp 1000/kWh

Golongan 2 tarif : Rp 2000/kWh

Syarat penghitungan: Minimum pembayaran adalah 100 kWh sedangkan untuk pemakaian 1000 kWh dan seterusnya tarifnya ditambah 10% dari total pembayaran.

Contoh:

Golongan 1, pemakaian 50 kWh, maka jumlah bayarnya 100.000

Golongan 2, pemakaian 150 kWh, maka jumlah bayarnya 300.000

Golongan 1, pemakaian 1000 kWh, maka jumlah bayarnya 1.100.000

18. (**nama program: FaktorBil**) Buatlah teks algoritma untuk menentukan faktor-faktor bilangan dari bilangan integer sembarang N ($N>0$) yang dimasukan melalui keyboard.

Contoh:

N=6, faktor bilangannya adalah 1, 2, 3, dan 6

N=10, faktor bilangannya adalah 1, 2, 5, dan 10

19. (**nama program: CekPrima**) Buatlah teks algoritma untuk mengetahui sebuah bilangan integer sembarang N ($N>0$) yang dimasukan melalui keyboard termasuk bilangan prima atau bukan. Bilangan prima adalah sebuah bilangan yang memiliki 2 faktor bilangan, yaitu 1 dan dirinya sendiri.

Contoh:

N=2 adalah bilangan prima

N=4 adalah bukan bilangan prima

20. (**nama program: CekSempurna**) Buatlah teks algoritma untuk mengetahui sebuah bilangan integer sembarang N ($N > 0$) yang dimasukan melalui keyboard termasuk bilangan sempurna atau bukan. Bilangan sempurna adalah sebuah bilangan yang jumlah faktor-faktor bilangannya kecuali faktor bilangan itu sendiri nilainya sebesar bilangan itu sendiri.

Contoh:

N=2 adalah bukan bilangan sempurna

N=6 adalah bilangan sempurna

21. (**nama program: DeretPrima**) Buatlah teks algoritma untuk mencetak bilangan prima yang dapat dibentuk dari bilangan integer sembarang N ($N > 0$) yang dimasukan melalui keyboard.

Contoh:

N=2 adalah bukan bilangan sempurna

N=6 adalah bilangan sempurna

22. (**nama program: SiputNaik**) Seekor siput yang berada pada sebuah lubang sedalam N meter (N dimasukan melalui keyboard yang nilai $N > 0$). Siput pada setiap pagi hari akan mengalami kenaikan sebesar 0.3 meter, sedangkan setiap malam hari akan mengalami penurunan sebesar 0.1 meter. Buatlah teks algoritma untuk mengetahui berapa hari waktu yang dibutuhkan oleh siput tersebut untuk mencapai ketinggian N meter.

23. (**nama program: Tetris**) Diberikan sebuah integer $N > 0$ yang dimasukan melalui keyboard, maka tampilkanlah dilayar susunan karakter '*' sesuai dengan susunan sebagai berikut:

Jika N = 5, maka dilayar akan ditampilkan:

```
*  --1
** --2
*** --3
**** --4
***** --5
```

24. (**nama program: Tetris1**) Diberikan sebuah integer $N > 0$ yang dimasukan melalui keyboard, maka tampilkanlah dilayar susunan karakter '*' sesuai dengan susunan sebagai berikut:

Jika N = 5, maka dilayar akan ditampilkan:

```
***** --5
```

```

**** --4
*** --3
** --2
* --1

```

25. (**nama program: Tetris2**) Diberikan sebuah integer $N > 0$ yang dimasukan melalui keyboard, maka tampilkanlah dilayar susunan karakter `*` sesuai dengan susunan sebagai berikut:

Jika $N = 5$, maka dilayar akan ditampilkan:

```

* --1
** --2
*** --3
**** --4
***** --5
**** --4
*** --3
** --2
* --1

```

26. (**nama program: Tetris3**) Diberikan sebuah integer $N > 0$ yang dimasukan melalui keyboard, maka tampilkanlah dilayar susunan karakter `*` sesuai dengan susunan sebagai berikut:

Jika $N = 5$, maka dilayar akan ditampilkan: (Catatan deret $Un = 1/2 * n * (n+1)$)

Banyaknya $N = 5$

```

* --1
*** --3
***** --6
***** --10
***** --15

```

27. (**nama program: Segitiga**) Diberikan sebuah integer $N > 0$ yang dimasukan melalui keyboard, maka tampilkanlah dilayar susunan karakter `*` sesuai dengan susunan sebagai berikut:

Jika $N = 5$, maka dilayar akan ditampilkan:

```

* * * * * --5
* * * * --4
* * * --3
* * --2
* --1

```

28. (**nama program: JumDeret1**) Buatlah teks algoritma jika diberikan deret bilangan yang dapat dibentuk dari N integer sembarang ($N > 0$) adalah sebagai berikut: $1 + 2 + 3 + 4 + 5 + \dots + N$, maka hitunglah jumlah total deret bilangan tersebut.
29. (**nama program: JumDeret2**) Buatlah teks algoritma jika diberikan deret bilangan yang dapat dibentuk dari N integer sembarang ($N > 0$) adalah sebagai berikut: $1 + 3 + 5 + \dots + (2N-1)$, maka hitunglah jumlah total deret bilangan tersebut.
30. (**nama program: JumDeret3**) Buatlah teks algoritma jika diberikan deret bilangan yang dapat dibentuk dari N integer sembarang ($N > 0$) adalah sebagai berikut: $1 - 2 + 3 - 4 + 5 - \dots + N$, maka hitunglah jumlah total deret bilangan tersebut.
31. (**nama program: JumDeret11**) Soal nomor 11 selesaikanlah dengan menggunakan array/tabel, kemudian bandingkan waktu eksekusinya dengan solusi tanpa array.
32. (**nama program: JumDeret21**) Soal nomor 12 selesaikanlah dengan menggunakan array/tabel, kemudian bandingkan waktu eksekusinya dengan solusi tanpa array.
33. (**nama program: JumDeret31**) Soal nomor 13 selesaikanlah dengan menggunakan array/tabel, kemudian bandingkan waktu eksekusinya dengan solusi tanpa array.
34. (**nama program: DNAKambing**) Diberikan kumpulan DNA kambing dalam bilangan integer sembarang yang sudah terurut ke atas. Jika selisih DNA < 3 , maka kambing tersebut bersaudara, dan jika selisih DNA ≥ 3 , maka kambing tersebut bukan saudara. Buatlah teks algoritma untuk menentukan berapa banyak saudara dan tidak saudara dari kumpulan DNA kambing di atas.
35. (**nama program: KelBil**) Buatlah teks algoritma untuk menentukan kelipatan bilangan terkecil, jika diberikan sebuah array integer A sebarang.
- Example:**
 Array bil = 2 3 6 \rightarrow kelipatan terkecil nya adalah 6
 Array bil = 2 6 12 \rightarrow kelipatan terkecil nya adalah 12
 Array bil = 2 4 5 \rightarrow kelipatan terkecil nya adalah 20
 Array bil = 2 4 7 \rightarrow kelipatan terkecil nya adalah 28
36. (**nama program: IsiMatrik**) Buatlah teks algoritma untuk mengisi nilai pada elemen-elemen matrik yang memiliki ordo sama ($M \times N$) dengan aturan bahwa nilai 1 diisikan pada elemen-elemen diagonal utama, nilai 0 diisikan pada elemen-elemen di atas diagonal utama, dan nilai 2 diisikan pada elemen-elemen di bawah diagonal utama. Nilai M dan N integer sembarang ($M > 0$ dan $N > 0$) yang dimasukkan melalui keyboard. Nilai M dan N harus bernilai sama, jika tidak sama maka akan menampilkan pesan "Nilai M dan N harus sama".

Ilustrasi:

1	0	0	0
2	1	0	0
2	2	1	0
2	2	2	1

37. (**nama program: Kuadran**) Diberikan P adalah sebuah **Titik** yang memiliki elemen X dan Y sebuah integer sembarang, maka buatlah sebuah teks algoritma untuk melakukan pengecekan mengenai kuadran sebuah titik P yang dimasukan melalui keyboard.

Example

P(1,1), maka Kuadran I

P(-11,1), maka Kuadran II

P(-1,-2), maka Kuadran III

38. (**nama program: JumDere41t**) Buatlah teks algoritma (perhatikan skema pemrosesan sekuensialnya), jika diberikan N bilangan integer sembarang > 0 yang dimasukan melalui keyboard, maka akan menghasilkan sebuah deret bilangan 1, 3, 6, 10, 15, 21, ..., $\frac{1}{2} n (n + 1)$ dan jumlahan dari elemen deret yang dapat terbentuk sebanyak N buah elemen tersebut.

Example:

N=-3, maka akan menampilkan pesan "Tidak ada Deret Bilangan, sehingga $S_n=0$ "

N=0, maka akan menampilkan pesan "Tidak ada Deret Bilangan, sehingga $S_n=0$ "

N=3, maka akan menampilkan deret 1, 3, 6 dengan $S_n= 1+3+ 6 = 10$

N=6, maka akan menampilkan deret 1, 3, 6, 10, 15, 21 dengan $S_n=1+3+6+10+15+21=56$

39. (**nama program: CekPrima**) Buatlah teks algoritma, jika diberikan i bilangan integer sembarang > 0 yang dimasukan melalui keyboard, kemudian selidiki apakah bilangan i tersebut termasuk bilangan PRIMA atau BUKAN?

Bilangan Prima adalah sebuah bilangan yang habis dibagi dengan 1 dan bilangan itu sendiri (atau memiliki 2 faktor bilangan)

Example:

i=4 output: 4 BUKAN Bilangan Prima

i=2 output: 2 Bilangan Prima

40. (**nama program: CekSempurna**) Buatlah teks algoritma, jika diberikan i bilangan integer sembarang > 0 yang dimasukan melalui keyboard, kemudian selidiki apakah bilangan i tersebut termasuk bilangan SEMPURNA atau BUKAN?

Bilangan Sempurna adalah sebuah bilangan yang jumlahan faktor bilangan sama dengan bilangan itu sendiri, kecuali faktor bilangan itu sendiri)

Example:

i=1 output: 4 BUKAN Bilangan Sempurna

i=4 output: 4 BUKAN Bilangan Sempurna

i=2 output: 2 BUKAN Bilangan Sempurna

i=6 output: 6 Bilangan Sempurna

41. (**nama program: JumPrima**) Buatlah teks algoritma untuk menampilkan bilangan **Prima** yang dapat dibentuk dari bilangan integer N sembarang > 0 dan tampilkan juga hasil penjumlahan dari elemen bilangan prima yang terbentuk dari deret tersebut !

Bilangan Prima adalah sebuah bilangan memiliki 2 faktor bilangan (habis dibagi), yaitu bilangan 1 dan bilangan itu sendiri.

Example:

N=-1 output: Menampilkan pesan “Tidak Ada Bilangan Prima, sehingga Jumlahannya Sn=0”

N=0 output: Menampilkan pesan “Tidak Ada Bilangan Prima, sehingga Jumlahannya Sn=0”

N=6 output: 2+3+5 dengan Jumlahan Deret Bilangannya adalah Sn=10

N=7 output: 2+3+5+7 dengan Jumlahan Deret Bilangannya adalah Sn=17

42. (**nama program: BilPrimaN**) Buatlah algoritma untuk mencetak bilangan Prima sampai dengan bilangan integer sembarang N.

Example:

N=5 → Bilangan Primanya adalah 2, 3, 5

N=7 → Bilangan Primanya adalah 2, 3, 5, 7

N=14 → Bilangan Primanya adalah 2, 3, 5, 7, 11, 13

43. (**nama sub program : Faktorial**) Buatlah sebuah teks algoritma dengan menggunakan sebuah sub program (**Fungsi/Prosedur**) untuk menghitung nilai Faktorial dari sebuah bilangan integer N sembarang.

Example:

Faktorial(5)=5x4x3x2x1=120

Faktorial(8)=8x7x6x5x4x3x2x1

Faktorial(4)=4x3x2x1=24

Faktorial(N)=Nx(N-1)x(N-2)x ... x1

44. (**nama sub program: Pangkat**) Buatlah teks algoritma dalam bentuk sub program, jika diberikan i dan j bilangan integer sembarang, maka akan menuliskan nilai hasil pemangkatan kedua bilangan tersebut.

Example → Pangkat(i,j)

Pangkat(2,3) artinya 2x2x2 dan menghasilkan nilai 8

Pangkat(0,3) artinya $0 \times 0 \times 0$ dan menghasilkan nilai 0

Pangkat(2,-3) artinya $1/(2 \times 2 \times 2)$ dan menghasilkan nilai $1/8$

Pangkat(2,0) menghasilkan nilai 1

Pangkat(-2,0) menghasilkan nilai 1

Pangkat(0,0) menghasilkan nilai 1

45. Diberikan sebuah konstanta integer Nmin=1 dan dan Nmax=100.

- a. **Procedure** CreateEmpty(T : array [Nmin . . Nmax])
{IS: Sembarang}
{FS: menghasilkan sebuah tabel T yang diinisialisasi dengan setiap elemennya bernilai 0}
- b. **Function** IsEmpty(T : array [Nmin . . Nmax]) → Boolean
{Mengirimkan True jika semua nilai elemennya bernilai 0}
- c. **Function** IsFull(T : array [Nmin . . Nmax]) → Boolean
{Mengirimkan True jika semua (sama dengan Nmax) nilai elemennya bernilai 1}
- d. **Function** NBElement(T : array [Nmin . . Nmax]) → integer
{Mengirimkan banyaknya elemen tabel T yang bernilai 1}
- e. **Procedure** SetElmtTabel(T : array [Nmin . . Nmax], N:integer)
{IS: Tabel masih Kosong semua elemen bernilai 0}
{FS: Tabel T terisi untuk setiap elemen sebanyak N dengan nilai 1}

46. Next ...