

100 Soal Pemrograman C

Lima materi dasar, 20 soal per materi, lengkap dengan kunci jawaban dan pembahasan. Setiap kode solusi sudah dikompilasi dengan GCC dan diuji terhadap contoh input-output di soalnya.

Bab 1	Pengenalan Algoritma dan Pemrograman C I	Soal 1.01 – 1.20
Bab 2	Pengenalan Pemrograman C II & Input/Output	Soal 2.01 – 2.20
Bab 3	Kontrol Program: Seleksi	Soal 3.01 – 3.20
Bab 4	Operator, Operan, dan Aritmatika	Soal 4.01 – 4.20
Bab 5	Kontrol Program: Perulangan	Soal 5.01 – 5.20
—	Kunci Jawaban dan Pembahasan	100 solusi

Cara Memakai Kumpulan Soal Ini

- Setiap soal mengikuti aturan Online Judge: program membaca dari **standard input** dan menulis ke **standard output**.
- Keluaran harus **persis sama** dengan format yang diminta, termasuk spasi, huruf besar-kecil, dan baris baru. Jangan mencetak teks tambahan seperti `Masukkan angka:` .
- Batasan (constraints) menjamin input selalu valid. Gunakan batasan untuk memilih tipe data yang cukup, misalnya `long long` jika hasilnya bisa melebihi sekitar $2,1 \times 10^9$.
- Program yang memakai `math.h` di GCC dikompilasi dengan opsi `-lm`, contohnya `gcc soal.c -o soal -lm`.
- Soal hanya memakai materi pada bab tersebut dan bab-bab sebelumnya. Materi Pointer dan Array tidak dipakai.
- Kerjakan dulu sendiri, baru cocokkan dengan kunci jawaban di bagian akhir. Solusi lain yang menghasilkan keluaran sama juga benar.

Daftar Soal

Bab 1 · Pengenalan Algoritma dan Pemrograman C I

1.01	Halo, Binusian!
1.02	Kartu Identitas
1.03	Luas dan Keliling Persegi Panjang
1.04	Konversi Detik
1.05	Rata-rata Tiga Bilangan
1.06	Kode ASCII
1.07	Karakter dari Kode
1.08	Huruf Berikutnya
1.09	Huruf Kapital
1.10	Celsius ke Fahrenheit
1.11	Akar Kuadrat
1.12	Pangkat
1.13	Volume Balok
1.14	Jarak Dua Titik
1.15	Tukar Nilai
1.16	Nilai Mutlak
1.17	Total Belanja dan Pajak
1.18	Sisi Miring Segitiga Siku-siku
1.19	Akar Persamaan Kuadrat
1.20	Durasi Kegiatan

Bab 4 · Operator, Operan, dan Aritmatika

4.01	Lima Operasi Dasar
4.02	Jumlah Digit
4.03	Balik Tiga Digit
4.04	Pecahan Uang
4.05	Increment dan Decrement
4.06	Operator Gabungan
4.07	Operator Relasional
4.08	Operator Logika
4.09	AND, OR, XOR
4.10	Komplemen Satu
4.11	Geser Bit
4.12	Ganjil Genap dengan Bitwise
4.13	Pangkat Dua
4.14	Tukar dengan XOR
4.15	Bit ke-K
4.16	Maksimum dengan Ternary
4.17	Prioritas Operator
4.18	Hari, Jam, dan Menit
4.19	Pembulatan Rata-rata
4.20	Kelipatan Terdekat

Bab 2 · Pengenalan Pemrograman C II & Input/Output

2.01	Nilai Akhir Mata Kuliah
2.02	Pembagian Bulat dan Pecahan
2.03	Bagian Bulat dan Pecahan
2.04	Lingkaran dengan Konstanta
2.05	Ukuran Tipe Data
2.06	Kolom Rata Kanan
2.07	Label Harga
2.08	Sapaan Nama Lengkap
2.09	Format Tanggal
2.10	Menit Sejak Tengah Malam
2.11	Oktal dan Heksadesimal
2.12	Membaca Heksadesimal
2.13	Dua Karakter Terbalik
2.14	Singkatan Tiga Huruf
2.15	Perkalian Besar
2.16	Meluap di Unsigned Char
2.17	Struk Belanja
2.18	Kalimat dalam Tanda Kutip
2.19	Label Diskon
2.20	Angka dari Karakter

Bab 5 · Kontrol Program: Perulangan

5.01	Hitung Sampai N
5.02	Jumlah 1 sampai N
5.03	Faktorial
5.04	Bilangan Ganjil dalam Rentang
5.05	Tabel Perkalian
5.06	Segitiga Bintang
5.07	Piramida Bintang
5.08	Jumlah Digit Bilangan Besar
5.09	Membalik Bilangan
5.10	Bilangan Prima
5.11	FPB dan KPK
5.12	Deret Fibonacci
5.13	Rata-rata dengan Sentinel
5.14	Desimal ke Biner
5.15	Bilangan Sempurna
5.16	Langkah Collatz
5.17	Jumlah dengan Continue
5.18	Tebak Angka
5.19	Bilangan Armstrong
5.20	Persegi Berongga

Bab 3 · Kontrol Program: Seleksi

3.01	Ganjil atau Genap
3.02	Positif, Negatif, atau Nol
3.03	Terbesar dari Tiga
3.04	Tahun Kabisat
3.05	Nilai Huruf
3.06	Predikat Kelulusan
3.07	Kalkulator Sederhana
3.08	Nama Hari
3.09	Jenis Segitiga
3.10	Jenis Akar Persamaan Kuadrat
3.11	Habis Dibagi 3 atau 5
3.12	Tarif Parkir
3.13	Kuadran Titik
3.14	Diskon Bertingkat
3.15	Vokal atau Konsonan
3.16	Jumlah Hari dalam Sebulan
3.17	Indeks Massa Tubuh
3.18	Urutkan Tiga Bilangan
3.19	Tagihan Listrik Bertingkat
3.20	Format Jam 12

Pengenalan Algoritma dan Pemrograman C I

Soal-soal di bab ini hanya memakai struktur sekuensial: membaca input, menghitung, lalu mencetak hasil. Belum perlu percabangan maupun perulangan. Fokusnya pada struktur program C, fungsi main, karakter ASCII, serta pustaka standar seperti math.h dan stdlib.h.

Soal 1.01 Halo, Binusian!

[Lihat jawaban →](#)

Program pertama setiap pemrogram adalah program yang menyapa dunia. Buatlah program yang mencetak kalimat `Hello, Binusian!` ke layar.

FORMAT INPUT

Tidak ada input.

FORMAT OUTPUT

Satu baris berisi teks `Hello, Binusian!`.

BATASAN

- Tidak ada.

Contoh Input 1

(tidak ada input)

Contoh Output 1

Hello, Binusian!

Soal 1.02 Kartu Identitas

[Lihat jawaban →](#)

Panitia ospek ingin mencetak kartu identitas sederhana. Diberikan nama panggilan (satu kata) dan umur seorang mahasiswa, cetak kartu identitasnya dengan format yang ditentukan.

FORMAT INPUT

Satu baris berisi sebuah kata `S` (nama panggilan) dan bilangan bulat `U` (umur), dipisahkan spasi.

FORMAT OUTPUT

Dua baris: `Nama: S` lalu `Umur: U tahun`.

BATASAN

- Panjang `S` 1 sampai 20 karakter, hanya huruf.
- $1 \leq U \leq 100$

Contoh Input 1

Budi 18

Contoh Output 1

Nama: Budi
Umur: 18 tahun

Soal 1.03 Luas dan Keliling Persegi Panjang[Lihat jawaban →](#)

Diberikan panjang dan lebar sebuah persegi panjang. Hitung luas dan kelilingnya.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `P` dan `L`.

FORMAT OUTPUT

Dua baris: `Luas = X` dan `Keliling = Y`.

BATASAN

- $1 \leq P, L \leq 10.000$

Contoh Input 1

5 3

Contoh Output 1

Luas = 15
Keliling = 16

Soal 1.04 Konversi Detik[Lihat jawaban →](#)

Sebuah stopwatch hanya menampilkan jumlah detik. Ubah jumlah detik tersebut menjadi jam, menit, dan detik.

FORMAT INPUT

Satu bilangan bulat `S` (jumlah detik).

FORMAT OUTPUT

Satu baris dengan format `H jam M menit D detik`.

BATASAN

- $0 \leq S \leq 1.000.000$

Contoh Input 1

3725

Contoh Output 1

1 jam 2 menit 5 detik

Contoh Input 2

59

Contoh Output 2

0 jam 0 menit 59 detik

Soal 1.05 Rata-rata Tiga Bilangan[Lihat jawaban →](#)

Hitung rata-rata dari tiga bilangan bulat yang diberikan.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `A`, `B`, dan `C`.

FORMAT OUTPUT

Rata-ratanya dengan tepat 2 angka di belakang koma.

BATASAN

- $-1.000.000 \leq A, B, C \leq 1.000.000$

Contoh Input 1

70 85 90

Contoh Output 1

81.67

Soal 1.06 Kode ASCII[Lihat jawaban →](#)

Setiap karakter di komputer disimpan sebagai angka menurut tabel ASCII. Diberikan sebuah karakter, cetak kode ASCII-nya.

FORMAT INPUT

Satu karakter yang bukan spasi.

FORMAT OUTPUT

Kode ASCII dari .

BATASAN

- adalah karakter ASCII yang dapat dicetak (kode 33 sampai 126).

Contoh Input 1

Contoh Output 1

Contoh Input 2

Contoh Output 2

Soal 1.07 Karakter dari Kode[Lihat jawaban →](#)

Kebalikan dari soal sebelumnya: diberikan sebuah kode ASCII, cetak karakter yang diwakilinya.

FORMAT INPUT

Satu bilangan bulat .

FORMAT OUTPUT

Satu karakter yang memiliki kode ASCII .

BATASAN

- $33 \leq N \leq 126$

Contoh Input 1

Contoh Output 1

Contoh Input 2

Contoh Output 2

Soal 1.08 Huruf Berikutnya[Lihat jawaban →](#)

Diberikan sebuah huruf kecil selain `z`. Cetak huruf yang tepat berada setelahnya dalam alfabet.

FORMAT INPUT

Satu huruf kecil `C`.

FORMAT OUTPUT

Huruf setelah `C`.

BATASAN

- `C` adalah huruf `a` sampai `y`.

Contoh Input 1

a

Contoh Output 1

b

Contoh Input 2

m

Contoh Output 2

n

Soal 1.09 Huruf Kapital[Lihat jawaban →](#)

Ubah sebuah huruf kecil menjadi huruf kapital tanpa memakai fungsi pustaka.

FORMAT INPUT

Satu huruf kecil `C`.

FORMAT OUTPUT

Huruf kapital dari `C`.

BATASAN

- `C` adalah huruf `a` sampai `z`.

Contoh Input 1

g

Contoh Output 1

G

Soal 1.10 Celsius ke Fahrenheit[Lihat jawaban →](#)

Ubah suhu dari derajat Celsius ke derajat Fahrenheit dengan rumus $F = C \times 9 / 5 + 32$.

FORMAT INPUT

Satu bilangan bulat `C`.

FORMAT OUTPUT

Suhu dalam Fahrenheit dengan 2 angka di belakang koma.

BATASAN

- $-273 \leq C \leq 1000$

Contoh Input 1

37

Contoh Output 1

98.60

Contoh Input 2

-40

Contoh Output 2

-40.00

Soal 1.11 Akar Kuadrat[Lihat jawaban →](#)

Gunakan pustaka `math.h` untuk menghitung akar kuadrat dari sebuah bilangan.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Akar kuadrat dari `N` dengan 3 angka di belakang koma.

BATASAN

- $0 \leq N \leq 1.000.000.000$

Contoh Input 1

2

Contoh Output 1

1.414

Contoh Input 2

144

Contoh Output 2

12.000

Soal 1.12 Pangkat[Lihat jawaban →](#)

Hitung A pangkat B dengan fungsi `pow` dari `math.h`.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Nilai A pangkat B sebagai bilangan bulat.

BATASAN

- $1 \leq A \leq 10$
- $0 \leq B \leq 9$

Contoh Input 1

2 10

Contoh Output 1

1024

Contoh Input 2

7 0

Contoh Output 2

1

Soal 1.13 Volume Balok[Lihat jawaban →](#)

Diberikan panjang, lebar, dan tinggi sebuah balok. Hitung volume dan luas permukaannya.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `P`, `L`, dan `T`.

BATASAN

- $1 \leq P, L, T \leq 1.000$

Contoh Input 1

2 3 4

FORMAT OUTPUT

Dua baris: `Volume = V` dan `Luas Permukaan = S`.

Contoh Output 1

Volume = 24
Luas Permukaan = 52

Soal 1.14 Jarak Dua Titik[Lihat jawaban →](#)

Hitung jarak antara titik $(X1, Y1)$ dan $(X2, Y2)$ pada bidang kartesius.

FORMAT INPUT

Satu baris berisi empat bilangan bulat `X1 Y1 X2 Y2`.

BATASAN

- $-10.000 \leq X1, Y1, X2, Y2 \leq 10.000$

Contoh Input 1

0 0 3 4

Contoh Input 2

1 1 2 2

FORMAT OUTPUT

Jarak kedua titik dengan 2 angka di belakang koma.

Contoh Output 1

5.00

Contoh Output 2

1.41

Soal 1.15 Tukar Nilai[Lihat jawaban →](#)

Diberikan dua bilangan `A` dan `B`. Tukar isi kedua variabel tersebut menggunakan variabel bantu, lalu cetak `A` dan `B` setelah ditukar.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Satu baris berisi nilai `A` dan `B` setelah ditukar, dipisahkan spasi.

BATASAN

- $-10^9 \leq A, B \leq 10^9$

Contoh Input 1

3 8

Contoh Output 1

8 3

Soal 1.16 Nilai Mutlak[Lihat jawaban →](#)

Cetak nilai mutlak dari sebuah bilangan bulat menggunakan fungsi `abs` dari `stdlib.h`.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Nilai mutlak dari `N`.

BATASAN

- $-10^9 \leq N \leq 10^9$

Contoh Input 1

-17

Contoh Output 1

17

Contoh Input 2

5

Contoh Output 2

5

Soal 1.17 Total Belanja dan Pajak[Lihat jawaban →](#)

Sebuah toko mengenakan pajak 10% dari total belanja. Diberikan harga satuan dan jumlah barang, cetak total sebelum pajak, besar pajak, dan total yang harus dibayar.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **H** (harga satuan) dan **N** (jumlah barang).

BATASAN

- $10 \leq H \leq 1.000.000$, dan **H** kelipatan 10
- $1 \leq N \leq 100$

Contoh Input 1

15000 3

FORMAT OUTPUT

Tiga baris: Subtotal: **X**, Pajak: **Y**, Total: **Z**.

Contoh Output 1

Subtotal: 45000
Pajak: 4500
Total: 49500

Soal 1.18 Sisi Miring Segitiga Siku-siku[Lihat jawaban →](#)

Diberikan panjang dua sisi tegak **A** dan **B** sebuah segitiga siku-siku. Hitung panjang sisi miring, keliling, dan luasnya.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **A** dan **B**.

BATASAN

- $1 \leq A, B \leq 10.000$

Contoh Input 1

3 4

Contoh Input 2

1 2

FORMAT OUTPUT

Tiga baris berisi sisi miring, keliling, dan luas, masing-masing dengan 2 angka di belakang koma.

Contoh Output 1

5.00
12.00
6.00

Contoh Output 2

2.24
5.24
1.00

Soal 1.19 Akar Persamaan Kuadrat[Lihat jawaban →](#)

Diberikan persamaan kuadrat $ax^2 + bx + c = 0$ yang dijamin memiliki dua akar real berbeda (diskriminan positif). Hitung kedua akarnya dengan rumus abc.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `a`, `b`, dan `c`.

BATASAN

- $1 \leq a \leq 100$
- $-1.000 \leq b, c \leq 1.000$
- $b^2 - 4ac > 0$

Contoh Input 1

```
1 -3 2
```

Contoh Input 2

```
2 1 -1
```

FORMAT OUTPUT

Dua baris: akar yang lebih besar, lalu akar yang lebih kecil, masing-masing dengan 2 angka di belakang koma.

Contoh Output 1

```
2.00  
1.00
```

Contoh Output 2

```
0.50  
-1.00
```

Soal 1.20 Durasi Kegiatan[Lihat jawaban →](#)

Sebuah kegiatan dimulai pada jam `H1:M1:S1` dan selesai pada jam `H2:M2:S2` di hari yang sama. Hitung lama kegiatan dalam detik.

FORMAT INPUT

Dua baris. Baris pertama berisi `H1 M1 S1`, baris kedua berisi `H2 M2 S2`.

BATASAN

- $0 \leq H \leq 23, 0 \leq M, S \leq 59$
- Waktu selesai tidak lebih awal dari waktu mulai.

Contoh Input 1

```
08 30 00  
10 15 30
```

Contoh Output 1

```
6330
```

Pengenalan Pemrograman C II & Input/Output

Bab ini melatih tipe data, casting, konstanta, sizeof, serta operasi input/output terformat: lebar kolom, presisi, rata kiri, membaca format khusus dengan scanf, getchar, dan membaca satu baris penuh.

Soal 2.01 Nilai Akhir Mata Kuliah

[Lihat jawaban →](#)

Nilai akhir sebuah mata kuliah dihitung dari 20% nilai tugas, 30% nilai UTS, dan 50% nilai UAS. Hitung nilai akhir seorang mahasiswa.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat: nilai tugas `T`, nilai UTS `M`, dan nilai UAS `F`.

FORMAT OUTPUT

Nilai akhir dengan 2 angka di belakang koma.

BATASAN

- $0 \leq T, M, F \leq 100$

Contoh Input 1

80 70 90

Contoh Output 1

82.00

Contoh Input 2

75 68 81

Contoh Output 2

75.90

Soal 2.02 Pembagian Bulat dan Pecahan[Lihat jawaban →](#)

Di C, hasil pembagian dua `int` selalu dibulatkan ke bawah (untuk bilangan positif). Tunjukkan perbedaannya: cetak hasil A / B sebagai bilangan bulat, lalu sebagai bilangan pecahan dengan casting.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Dua baris: hasil pembagian bulat, lalu hasil pembagian pecahan dengan 3 angka di belakang koma.

BATASAN

- $0 \leq A \leq 10^9$
- $1 \leq B \leq 10^9$

Contoh Input 1

7 2

Contoh Output 1

3
3.500

Contoh Input 2

10 3

Contoh Output 2

3
3.333

Soal 2.03 Bagian Bulat dan Pecahan[Lihat jawaban →](#)

Diberikan sebuah bilangan real positif `X`. Pisahkan bagian bulat dan bagian pecahannya menggunakan casting.

FORMAT INPUT

Satu bilangan real `X` dengan paling banyak 2 angka di belakang koma.

FORMAT OUTPUT

Dua baris: bagian bulat `X`, lalu bagian pecahannya dengan 2 angka di belakang koma.

BATASAN

- $0 \leq X < 1.000.000$

Contoh Input 1

3.75

Contoh Output 1

3
0.75

Contoh Input 2

12.5

Contoh Output 2

12
0.50

Soal 2.04 Lingkaran dengan Konstanta[Lihat jawaban →](#)

Definisikan konstanta `PI` bernilai 3.14159 menggunakan `#define`, lalu hitung luas dan keliling lingkaran berjari-jari `R`.

FORMAT INPUT

Satu bilangan bulat `R`.

FORMAT OUTPUT

Dua baris: luas dan keliling, masing-masing dengan 3 angka di belakang koma.

BATASAN

- $1 \leq R \leq 10.000$

Contoh Input 1

7

Contoh Output 1

153.938
43.982

Soal 2.05 Ukuran Tipe Data[Lihat jawaban →](#)

Gunakan operator `sizeof` untuk mencetak ukuran (dalam byte) dari tipe `char`, `int`, `float`, dan `double`. Anggap kompiler yang dipakai adalah GCC 32/64-bit standar.

FORMAT INPUT

Tidak ada input.

FORMAT OUTPUT

Empat baris dengan format `tipe: ukuran`.

BATASAN

- Tidak ada.

Contoh Input 1

(tidak ada input)

Contoh Output 1

char: 1
int: 4
float: 4
double: 8

Soal 2.06 Kolom Rata Kanan[Lihat jawaban →](#)

Cetak tiga bilangan bulat, masing-masing di barisnya sendiri dan rata kanan dalam kolom selebar 8 karakter.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `A`, `B`, dan `C`.

BATASAN

- $-9.999.999 \leq A, B, C \leq 99.999.999$

Contoh Input 1

```
5 -120 3000
```

FORMAT OUTPUT

Tiga baris; tiap bilangan rata kanan dengan lebar 8 (diisi spasi di kiri).

Contoh Output 1

```
      5
     -120
     3000
```

Soal 2.07 Label Harga[Lihat jawaban →](#)

Cetak nama barang rata kiri selebar 10 karakter dan harganya rata kanan selebar 10 karakter dengan 2 angka di belakang koma, diapit tanda `|`.

FORMAT INPUT

Satu baris berisi nama barang `S` (satu kata) dan harga `H` (bilangan real).

BATASAN

- Panjang `S` 1 sampai 10 karakter.
- $0 \leq H < 1.000.000$

Contoh Input 1

```
Apel 12500.5
```

FORMAT OUTPUT

Satu baris dengan format `|S|H|` sesuai lebar yang ditentukan.

Contoh Output 1

```
|Apel      | 12500.50|
```

Soal 2.08 Sapaan Nama Lengkap[Lihat jawaban →](#)

Baca nama lengkap seseorang yang bisa mengandung spasi, lalu sapa dia.

FORMAT INPUT

Satu baris berisi nama lengkap `S`.

BATASAN

- Panjang `S` 1 sampai 100 karakter, hanya huruf dan spasi.

Contoh Input 1

```
Budi Santoso
```

FORMAT OUTPUT

```
Halo, S!
```

Contoh Output 1

```
Halo, Budi Santoso!
```

Soal 2.09 Format Tanggal[Lihat jawaban →](#)

Ubah tanggal dari format `DD/MM/YYYY` menjadi format `YYYY-MM-DD`.

FORMAT INPUT

Satu baris berisi tanggal dengan format `DD/MM/YYYY`.

FORMAT OUTPUT

Tanggal dalam format `YYYY-MM-DD`. Hari dan bulan selalu ditulis dua digit.

BATASAN

- Tanggal dijamin valid, tahun 1000 sampai 9999.

Contoh Input 1

17/08/1945

Contoh Output 1

1945-08-17

Contoh Input 2

1/1/2024

Contoh Output 2

2024-01-01

Soal 2.10 Menit Sejak Tengah Malam[Lihat jawaban →](#)

Diberikan jam dalam format `HH:MM`. Hitung berapa menit telah berlalu sejak pukul 00:00.

FORMAT INPUT

Satu baris berisi waktu `HH:MM`.

FORMAT OUTPUT

Jumlah menit sejak tengah malam.

BATASAN

- $00 \leq HH \leq 23$, $00 \leq MM \leq 59$

Contoh Input 1

13:45

Contoh Output 1

825

Contoh Input 2

00:07

Contoh Output 2

7

Soal 2.11 Oktal dan Heksadesimal[Lihat jawaban →](#)

Cetak sebuah bilangan bulat dalam basis 8 (oktal) dan basis 16 (heksadesimal huruf kecil dan huruf besar).

FORMAT INPUT

Satu bilangan bulat `N`.

BATASAN

- $0 \leq N \leq 10^9$

Contoh Input 1

255

Contoh Input 2

100

FORMAT OUTPUT

Tiga baris: `N` dalam oktal, heksadesimal huruf kecil, dan heksadesimal huruf besar.

Contoh Output 1

377
ff
FF

Contoh Output 2

144
64
64

Soal 2.12 Membaca Heksadesimal[Lihat jawaban →](#)

Diberikan sebuah bilangan dalam basis 16. Cetak nilainya dalam basis 10.

FORMAT INPUT

Satu bilangan heksadesimal `H` (huruf boleh besar atau kecil, tanpa awalan `0x`).

BATASAN

- $0 \leq H \leq 7FFFFFFF$

Contoh Input 1

1F

Contoh Input 2

ff

FORMAT OUTPUT

Nilai `H` dalam desimal.

Contoh Output 1

31

Contoh Output 2

255

Soal 2.13 Dua Karakter Terbalik[Lihat jawaban →](#)

Baca dua karakter yang diketik berdempetan menggunakan `getchar`, lalu cetak dengan urutan terbalik.

FORMAT INPUT

Satu baris berisi tepat dua karakter tanpa spasi di antaranya.

FORMAT OUTPUT

Kedua karakter dalam urutan terbalik.

BATASAN

- Kedua karakter adalah karakter ASCII yang dapat dicetak dan bukan spasi.

Contoh Input 1

AB

Contoh Output 1

BA

Contoh Input 2

7x

Contoh Output 2

x7

Soal 2.14 Singkatan Tiga Huruf[Lihat jawaban →](#)

Kode bandara dibuat dari tiga huruf pertama nama kota. Diberikan nama kota, cetak tiga huruf pertamanya menggunakan batas lebar pada `scanf`.

FORMAT INPUT

Satu kata `S`.

FORMAT OUTPUT

Tiga karakter pertama dari `S`.

BATASAN

- $3 \leq \text{panjang } S \leq 50$

Contoh Input 1

Jakarta

Contoh Output 1

Jak

Contoh Input 2

Bandung

Contoh Output 2

Ban

Soal 2.15 Perkalian Besar[Lihat jawaban →](#)

Hitung hasil kali dua bilangan yang bisa sangat besar. Gunakan tipe data yang cukup menampung hasilnya.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Hasil kali `A × B`.

BATASAN

- $-10^9 \leq A, B \leq 10^9$

Contoh Input 1

1000000000 999999999

Contoh Output 1

999999999000000000

Contoh Input 2

-3 7

Contoh Output 2

-21

Soal 2.16 Meluap di Unsigned Char[Lihat jawaban →](#)

Variabel `unsigned char` hanya menampung nilai 0 sampai 255. Apa yang terjadi jika nilai yang lebih besar dimasukkan? Cetak hasil casting `N` ke `unsigned char`.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Nilai `(unsigned char) N` sebagai bilangan bulat.

BATASAN

- $0 \leq N \leq 100.000$

Contoh Input 1

300

Contoh Output 1

44

Contoh Input 2

255

Contoh Output 2

255

Soal 2.17 Struk Belanja[Lihat jawaban →](#)

Cetak struk belanja dua barang dengan kolom yang rapi. Tiap baris barang berisi nama rata kiri selebar 10, jumlah rata kanan selebar 5, dan subtotal rata kanan selebar 12. Baris terakhir berisi **TOTAL** rata kiri selebar 15 dan total rata kanan selebar 12.

FORMAT INPUT

Dua baris. Tiap baris berisi nama barang (satu kata), jumlah **Q**, dan harga satuan **H**.

FORMAT OUTPUT

Tiga baris sesuai format di atas.

BATASAN

- Panjang nama 1 sampai 10 karakter.
- $1 \leq Q \leq 1.000$
- $1 \leq H \leq 1.000.000$

Contoh Input 1

```
Pensil 3 2500
Buku 2 12000
```

Contoh Output 1

```
Pensil      3      7500
Buku        2     24000
TOTAL                               31500
```

Soal 2.18 Kalimat dalam Tanda Kutip[Lihat jawaban →](#)

Baca satu baris kalimat, lalu cetak kalimat tersebut diapit tanda kutip ganda.

FORMAT INPUT

Satu baris berisi kalimat **S**.

FORMAT OUTPUT

Kalimat **S** diapit tanda **"**.

BATASAN

- Panjang **S** 1 sampai 200 karakter dan tidak mengandung tanda kutip.

Contoh Input 1

```
Hello World
```

Contoh Output 1

```
"Hello World"
```

Soal 2.19 Label Diskon[Lihat jawaban →](#)

Sebuah toko menampilkan label diskon. Diberikan harga awal dan persentase diskon, cetak besar diskon dan harga yang harus dibayar.

FORMAT INPUT

Satu baris berisi dua bilangan bulat: harga `H` dan diskon `D` (dalam persen).

BATASAN

- $100 \leq H \leq 10^8$, dan `H` kelipatan 100
- $0 \leq D \leq 100$

Contoh Input 1

100000 25

FORMAT OUTPUT

Dua baris: `Diskon D%: Rp X` dan `Bayar: Rp Y`.

Contoh Output 1

Diskon 25%: Rp 25000
Bayar: Rp 75000

Soal 2.20 Angka dari Karakter[Lihat jawaban →](#)

Baca sebuah digit sebagai karakter dengan `getchar`, ubah menjadi nilai angkanya, lalu cetak nilai tersebut beserta kuadratnya.

FORMAT INPUT

Satu karakter digit `C` (`0` sampai `9`).

BATASAN

- `C` adalah karakter `0` sampai `9`.

Contoh Input 1

7

Contoh Output 1

7 49

Contoh Input 2

0

Contoh Output 2

0 0

Kontrol Program: Seleksi

Bab ini melatih percabangan dengan if, if-else, else-if bertingkat, nested if, switch-case, dan operator ternary. Perhatikan urutan pengecekan kondisi: urutan yang salah adalah sumber logical error yang paling umum.

Soal 3.01 Ganjil atau Genap

[Lihat jawaban →](#)

Tentukan apakah sebuah bilangan bulat termasuk ganjil atau genap.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

`GENAP` jika `N` genap, `GANJIL` jika ganjil.

BATASAN

- $-10^9 \leq N \leq 10^9$

Contoh Input 1

4

Contoh Output 1

GENAP

Contoh Input 2

-7

Contoh Output 2

GANJIL

Soal 3.02 Positif, Negatif, atau Nol

[Lihat jawaban →](#)

Tentukan jenis sebuah bilangan bulat: positif, negatif, atau nol.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

`POSITIF`, `NEGATIF`, atau `NOL`.

BATASAN

- $-10^9 \leq N \leq 10^9$

Contoh Input 1

15

Contoh Output 1

POSITIF

Contoh Input 2

0

Contoh Output 2

NOL

Soal 3.03 Terbesar dari Tiga[Lihat jawaban →](#)

Diberikan tiga bilangan bulat. Cetak bilangan yang paling besar.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `A`, `B`, dan `C`.

FORMAT OUTPUT

Bilangan terbesar.

BATASAN

- $-10^9 \leq A, B, C \leq 10^9$

Contoh Input 1

3 9 5

Contoh Output 1

9

Contoh Input 2

-1 -8 -1

Contoh Output 2

-1

Soal 3.04 Tahun Kabisat[Lihat jawaban →](#)

Tahun kabisat adalah tahun yang habis dibagi 4 tetapi tidak habis dibagi 100, atau tahun yang habis dibagi 400. Tentukan apakah sebuah tahun adalah tahun kabisat.

FORMAT INPUT

Satu bilangan bulat `Y`.

FORMAT OUTPUT

`KABISAT` atau `BUKAN KABISAT`.

BATASAN

- $1 \leq Y \leq 10.000$

Contoh Input 1

2024

Contoh Output 1

KABISAT

Contoh Input 2

1900

Contoh Output 2

BUKAN KABISAT

Contoh Input 3

2000

Contoh Output 3

KABISAT

Soal 3.05 Nilai Huruf[Lihat jawaban →](#)

Ubah nilai angka menjadi nilai huruf dengan aturan: 85–100 = A, 70–84 = B, 60–69 = C, 50–59 = D, dan di bawah 50 = E.

FORMAT INPUT

Satu bilangan bulat **N** (nilai angka).

FORMAT OUTPUT

Satu huruf: nilai hurufnya.

BATASAN

- $0 \leq N \leq 100$

Contoh Input 1

88

Contoh Output 1

A

Contoh Input 2

69

Contoh Output 2

C

Contoh Input 3

12

Contoh Output 3

E

Soal 3.06 Predikat Kelulusan[Lihat jawaban →](#)

Tentukan predikat kelulusan berdasarkan IPK: 3,50–4,00 = **Cumlaude** ; 3,00–3,49 = **Sangat Memuaskan** ; 2,50–2,99 = **Memuaskan** ; 2,00–2,49 = **Cukup** ; di bawah 2,00 = **Kurang** .

FORMAT INPUT

Satu bilangan real **IPK** dengan tepat 2 angka di belakang koma.

FORMAT OUTPUT

Predikat kelulusan.

BATASAN

- $0.00 \leq \text{IPK} \leq 4.00$

Contoh Input 1

3.75

Contoh Output 1

Cumlaude

Contoh Input 2

2.99

Contoh Output 2

Memuaskan

Contoh Input 3

3.00

Contoh Output 3

Sangat Memuaskan

Soal 3.07 Kalkulator Sederhana[Lihat jawaban →](#)

Buat kalkulator untuk empat operasi dasar (+, -, *, /) pada bilangan bulat menggunakan `switch-case`. Pembagian adalah pembagian bulat. Jika terjadi pembagian dengan nol, cetak `Error`.

FORMAT INPUT

Satu baris berisi `A op B`, dengan `op` salah satu dari `+ - * /`.

BATASAN

- $-10^4 \leq A, B \leq 10^4$

FORMAT OUTPUT

Hasil operasi, atau `Error`.

Contoh Input 1

12 * 7

Contoh Output 1

84

Contoh Input 2

9 / 0

Contoh Output 2

Error

Contoh Input 3

17 / 5

Contoh Output 3

3

Soal 3.08 Nama Hari[Lihat jawaban →](#)

Ubah nomor hari menjadi nama hari: 1 = Senin, 2 = Selasa, sampai 7 = Minggu. Jika nomornya di luar 1 sampai 7, cetak `Tidak valid`.

FORMAT INPUT

Satu bilangan bulat `N`.

BATASAN

- $-100 \leq N \leq 100$

FORMAT OUTPUT

Nama hari, atau `Tidak valid`.

Contoh Input 1

3

Contoh Output 1

Rabu

Contoh Input 2

9

Contoh Output 2

Tidak valid

Soal 3.09 Jenis Segitiga[Lihat jawaban →](#)

Diberikan panjang tiga sisi. Tentukan apakah ketiganya bisa membentuk segitiga, dan jika bisa, jenisnya.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `A`, `B`, dan `C`.

BATASAN

- $1 \leq A, B, C \leq 10^9$

FORMAT OUTPUT

`Bukan Segitiga`, `Sama Sisi`, `Sama Kaki`, atau `Sembarang`.

Contoh Input 1

`5 5 5`

Contoh Input 2

`3 4 5`

Contoh Input 3

`1 2 3`

Contoh Input 4

`7 4 7`

Contoh Output 1

`Sama Sisi`

Contoh Output 2

`Sembarang`

Contoh Output 3

`Bukan Segitiga`

Contoh Output 4

`Sama Kaki`

Soal 3.10 Jenis Akar Persamaan Kuadrat[Lihat jawaban →](#)

Diberikan persamaan kuadrat $ax^2 + bx + c = 0$. Hitung diskriminannya dan tentukan akar-akarnya.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `a`, `b`, dan `c`.

FORMAT OUTPUT

Jika $D > 0$, cetak dua baris berisi akar terbesar lalu terkecil. Jika $D = 0$, cetak satu baris berisi akar kembaranya. Jika $D < 0$, cetak `Akar imajiner`. Semua akar dicetak dengan 2 angka di belakang koma.

BATASAN

- $1 \leq a \leq 100$
- $-1.000 \leq b, c \leq 1.000$

Contoh Input 1`1 -5 6`**Contoh Output 1**`3.00
2.00`**Contoh Input 2**`1 2 1`**Contoh Output 2**`-1.00`**Contoh Input 3**`1 0 4`**Contoh Output 3**`Akar imajiner`**Soal 3.11 Habis Dibagi 3 atau 5**[Lihat jawaban →](#)

Diberikan bilangan bulat positif `N`. Cetak `FizzBuzz` jika habis dibagi 3 dan 5, `Fizz` jika hanya habis dibagi 3, `Buzz` jika hanya habis dibagi 5, atau `N` itu sendiri jika tidak keduanya.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

`FizzBuzz`, `Fizz`, `Buzz`, atau nilai `N`.

BATASAN

- $1 \leq N \leq 10^9$

Contoh Input 1`15`**Contoh Output 1**`FizzBuzz`**Contoh Input 2**`9`**Contoh Output 2**`Fizz`**Contoh Input 3**`7`**Contoh Output 3**`7`

Soal 3.12 Tarif Parkir[Lihat jawaban →](#)

Tarif parkir sebuah mal: 2 jam pertama Rp5.000, lalu setiap jam berikutnya Rp3.000. Total tarif paling mahal Rp30.000. Hitung tarif parkir untuk lama parkir J jam.

FORMAT INPUT

Satu bilangan bulat J (lama parkir dalam jam, sudah dibulatkan ke atas).

FORMAT OUTPUT

Total tarif parkir.

BATASAN

- $1 \leq J \leq 1.000$

Contoh Input 1

1

Contoh Output 1

5000

Contoh Input 2

5

Contoh Output 2

14000

Contoh Input 3

24

Contoh Output 3

30000

Soal 3.13 Kuadran Titik[Lihat jawaban →](#)

Tentukan letak titik (X, Y) pada bidang kartesius.

FORMAT INPUT

Satu baris berisi dua bilangan bulat X dan Y .

FORMAT OUTPUT

Salah satu dari: Kuadran I, Kuadran II, Kuadran III, Kuadran IV, Sumbu X, Sumbu Y, atau Titik Asal.

BATASAN

- $-10^9 \leq X, Y \leq 10^9$

Contoh Input 1

3 -2

Contoh Output 1

Kuadran IV

Contoh Input 2

0 5

Contoh Output 2

Sumbu Y

Contoh Input 3

0 0

Contoh Output 3

Titik Asal

Soal 3.14 Diskon Bertingkat[Lihat jawaban →](#)

Sebuah toko memberi diskon 20% untuk belanja minimal Rp500.000, diskon 10% untuk belanja minimal Rp200.000, dan tanpa diskon untuk belanja di bawah itu. Hitung yang harus dibayar. Jika hasilnya pecahan, bulatkan ke bawah.

FORMAT INPUT

Satu bilangan bulat **T** (total belanja).

FORMAT OUTPUT

Jumlah yang harus dibayar.

BATASAN

- $1 \leq T \leq 10^9$

Contoh Input 1

600000

Contoh Output 1

480000

Contoh Input 2

250005

Contoh Output 2

225004

Contoh Input 3

199999

Contoh Output 3

199999

Soal 3.15 Vokal atau Konsonan[Lihat jawaban →](#)

Tentukan apakah sebuah karakter adalah huruf vokal, huruf konsonan, atau bukan huruf. Huruf besar dan kecil diperlakukan sama.

FORMAT INPUT

Satu karakter **C** yang bukan spasi.

FORMAT OUTPUT

Vokal, Konsonan, atau Bukan Huruf.

BATASAN

- **C** adalah karakter ASCII yang dapat dicetak.

Contoh Input 1

E

Contoh Output 1

Vokal

Contoh Input 2

k

Contoh Output 2

Konsonan

Contoh Input 3

7

Contoh Output 3

Bukan Huruf

Soal 3.16 Jumlah Hari dalam Sebulan[Lihat jawaban →](#)

Diberikan nomor bulan dan tahun. Cetak jumlah hari pada bulan tersebut, dengan memperhitungkan tahun kabisat untuk bulan Februari.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **M** (bulan) dan **Y** (tahun).

BATASAN

- $1 \leq M \leq 12$
- $1 \leq Y \leq 10.000$

FORMAT OUTPUT

Jumlah hari pada bulan tersebut.

Contoh Input 1

2 2024

Contoh Input 2

4 2023

Contoh Input 3

2 2100

Contoh Output 1

29

Contoh Output 2

30

Contoh Output 3

28

Soal 3.17 Indeks Massa Tubuh[Lihat jawaban →](#)

Hitung Indeks Massa Tubuh (IMT) = $\text{berat (kg)} / (\text{tinggi (m)})^2$, lalu tentukan kategorinya: di bawah 18,5 = **Kurus**; 18,5 sampai di bawah 25 = **Normal**; 25 sampai di bawah 30 = **Gemuk**; 30 ke atas = **Obesitas**.

FORMAT INPUT

Satu baris berisi dua bilangan bulat: berat **B** (kg) dan tinggi **T** (cm).

BATASAN

- $30 \leq B \leq 300$
- $100 \leq T \leq 250$

FORMAT OUTPUT

Satu baris berisi IMT dengan 1 angka di belakang koma, spasi, lalu kategorinya.

Contoh Input 1

70 175

Contoh Input 2

95 170

Contoh Output 1

22.9 Normal

Contoh Output 2

32.9 Obesitas

Soal 3.18 Urutkan Tiga Bilangan[Lihat jawaban →](#)

Urutkan tiga bilangan bulat dari yang terkecil ke yang terbesar hanya dengan percabangan (tanpa perulangan).

FORMAT INPUT

Satu baris berisi tiga bilangan bulat `A`, `B`, dan `C`.

FORMAT OUTPUT

Ketiga bilangan terurut naik, dipisahkan spasi.

BATASAN

- $-10^9 \leq A, B, C \leq 10^9$

Contoh Input 1

9 2 5

Contoh Output 1

2 5 9

Contoh Input 2

3 3 1

Contoh Output 2

1 3 3

Soal 3.19 Tagihan Listrik Bertingkat[Lihat jawaban →](#)

Tarif listrik dihitung bertingkat: 100 kWh pertama Rp1.000/kWh, 100 kWh berikutnya (kWh ke-101 sampai 200) Rp1.500/kWh, dan sisanya Rp2.000/kWh. Hitung total tagihan.

FORMAT INPUT

Satu bilangan bulat `K` (pemakaian dalam kWh).

FORMAT OUTPUT

Total tagihan.

BATASAN

- $0 \leq K \leq 100.000$

Contoh Input 1

80

Contoh Output 1

80000

Contoh Input 2

150

Contoh Output 2

175000

Contoh Input 3

250

Contoh Output 3

350000

Soal 3.20 Format Jam 12[Lihat jawaban →](#)

Ubah waktu dari format 24 jam `HH:MM` menjadi format 12 jam `hh:mm AM/PM`. Pukul 00:xx menjadi 12:xx AM, dan pukul 12:xx menjadi 12:xx PM.

FORMAT INPUT

Satu baris berisi waktu `HH:MM`.

FORMAT OUTPUT

Waktu dalam format `hh:mm AM` atau `hh:mm PM`, dengan jam dan menit selalu dua digit.

BATASAN

- $00 \leq HH \leq 23$, $00 \leq MM \leq 59$

Contoh Input 1`00:15`**Contoh Output 1**`12:15 AM`**Contoh Input 2**`13:05`**Contoh Output 2**`01:05 PM`**Contoh Input 3**`12:00`**Contoh Output 3**`12:00 PM`

Operator, Operan, dan Aritmatika

Bab ini melatih operator aritmetika (termasuk modulo), increment/decrement, operator gabungan, relasional, logika, bitwise, operator kondisional (ternary), serta prioritas operator. Sebagian besar soal bisa diselesaikan tanpa if, cukup dengan ekspresi yang tepat.

Soal 4.01 Lima Operasi Dasar

[Lihat jawaban →](#)

Diberikan dua bilangan bulat positif A dan B . Cetak hasil $A + B$, $A - B$, $A * B$, A / B (pembagian bulat), dan $A \% B$.

FORMAT INPUT

Satu baris berisi dua bilangan bulat A dan B .

BATASAN

- $1 \leq A, B \leq 10^4$

Contoh Input 1

17 5

FORMAT OUTPUT

Lima baris berisi kelima hasil operasi sesuai urutan di atas.

Contoh Output 1

22
12
85
3
2

Soal 4.02 Jumlah Digit

[Lihat jawaban →](#)

Diberikan bilangan tiga digit. Hitung jumlah ketiga digitnya tanpa perulangan.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $100 \leq N \leq 999$

Contoh Input 1

472

Contoh Input 2

100

FORMAT OUTPUT

Jumlah digit N .

Contoh Output 1

13

Contoh Output 2

1

Soal 4.03 Balik Tiga Digit[Lihat jawaban →](#)

Balik urutan digit sebuah bilangan tiga digit. Jika hasilnya diawali nol, cetak sebagai bilangan biasa (tanpa nol di depan).

FORMAT INPUT

Satu bilangan bulat `N`.

BATASAN

- $100 \leq N \leq 999$

Contoh Input 1

472

Contoh Input 2

120

FORMAT OUTPUT

Bilangan hasil pembalikan.

Contoh Output 1

274

Contoh Output 2

21

Soal 4.04 Pecahan Uang[Lihat jawaban →](#)

Seorang kasir ingin menyerahkan uang dengan jumlah lembar sesedikit mungkin. Pecahan yang tersedia: 50.000, 20.000, 10.000, 5.000, 2.000, dan 1.000. Hitung banyaknya lembar setiap pecahan.

FORMAT INPUT

Satu bilangan bulat `N` (jumlah uang).

BATASAN

- $0 \leq N \leq 10^9$, dan `N` kelipatan 1.000

Contoh Input 1

88000

Contoh Input 2

140000

FORMAT OUTPUT

Enam baris dengan format `pecahan: jumlah`, dari pecahan terbesar.

Contoh Output 1

50000: 1
20000: 1
10000: 1
5000: 1
2000: 1
1000: 1

Contoh Output 2

50000: 2
20000: 2
10000: 0
5000: 0
2000: 0
1000: 0

Soal 4.05 Increment dan Decrement[Lihat jawaban →](#)

Diberikan nilai awal `N`. Simulasikan potongan kode berikut dan cetak hasilnya: `printf("%d\n", ++n); printf("%d\n", n++); printf("%d\n", n);`

FORMAT INPUT

Satu bilangan bulat `N`.

BATASAN

- $-10^9 \leq N \leq 10^9$

Contoh Input 1

10

FORMAT OUTPUT

Tiga baris sesuai keluaran potongan kode.

Contoh Output 1

11
11
12

Soal 4.06 Operator Gabungan[Lihat jawaban →](#)

Diberikan `X` dan `Y`. Jalankan berurutan: `x *= y + 1;`, `x -= 3;`, `x /= 2;`, `x %= 7;`, lalu cetak nilai `x` setelah setiap langkah.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `X` dan `Y`.

BATASAN

- $1 \leq X, Y \leq 1.000$

Contoh Input 1

10 6

FORMAT OUTPUT

Empat baris berisi nilai `x` setelah setiap langkah.

Contoh Output 1

70
67
33
5

Soal 4.07 Operator Relasional[Lihat jawaban →](#)

Bandingkan dua bilangan `A` dan `B` dengan keenam operator relasional, lalu cetak hasilnya (1 untuk benar, 0 untuk salah).

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Enam baris berisi hasil `A == B`, `A != B`, `A < B`, `A > B`, `A <= B`, dan `A >= B`.

BATASAN

- $-10^9 \leq A, B \leq 10^9$

Contoh Input 1

5 6

Contoh Output 1

0
1
1
0
1
0

Soal 4.08 Operator Logika[Lihat jawaban →](#)

Dalam C, nilai 0 berarti salah dan nilai selain 0 berarti benar. Diberikan dua bilangan `A` dan `B`, cetak hasil `A && B`, `A || B`, `!A`, dan `!(A && B)`.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Empat baris berisi hasil keempat ekspresi (0 atau 1).

BATASAN

- $-10^9 \leq A, B \leq 10^9$

Contoh Input 1

23 0

Contoh Output 1

0
1
0
1

Contoh Input 2

-4 9

Contoh Output 2

1
1
0
0

Soal 4.09 AND, OR, XOR[Lihat jawaban →](#)

Hitung hasil operasi bitwise AND, OR, dan XOR dari dua bilangan bulat tak negatif.

FORMAT INPUT

Satu baris berisi dua bilangan bulat `A` dan `B`.

FORMAT OUTPUT

Tiga baris berisi `A & B`, `A | B`, dan `A ^ B`.

BATASAN

- $0 \leq A, B \leq 10^9$

Contoh Input 1

24 35

Contoh Output 1

0
59
59

Contoh Input 2

45 75

Contoh Output 2

9
111
102

Soal 4.10 Komplemen Satu[Lihat jawaban →](#)

Hitung komplemen satu (`~`) dari sebuah bilangan bulat bertipe `int`.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Nilai `~N`.

BATASAN

- $-10^9 \leq N \leq 10^9$

Contoh Input 1

5

Contoh Output 1

-6

Contoh Input 2

-1

Contoh Output 2

0

Soal 4.11 Geser Bit[Lihat jawaban →](#)

Diberikan N dan K . Cetak hasil geser kiri $N \ll K$ dan geser kanan $N \gg K$.

FORMAT INPUT

Satu baris berisi dua bilangan bulat N dan K .

FORMAT OUTPUT

Dua baris berisi $N \ll K$ dan $N \gg K$.

BATASAN

- $0 \leq N \leq 10^6$
- $0 \leq K \leq 10$

Contoh Input 1

78 2

Contoh Output 1

312
19

Contoh Input 2

78 3

Contoh Output 2

624
9

Soal 4.12 Ganjil Genap dengan Bitwise[Lihat jawaban →](#)

Tentukan apakah sebuah bilangan tak negatif ganjil atau genap tanpa operator $\%$ dan tanpa `if`.

FORMAT INPUT

Satu bilangan bulat N .

FORMAT OUTPUT

GANJIL atau GENAP.

BATASAN

- $0 \leq N \leq 10^9$

Contoh Input 1

13

Contoh Output 1

GANJIL

Contoh Input 2

64

Contoh Output 2

GENAP

Soal 4.13 Pangkat Dua[Lihat jawaban →](#)

Tentukan apakah sebuah bilangan bulat positif merupakan pangkat dari 2 (1, 2, 4, 8, ...), cukup dengan satu ekspresi bitwise.

FORMAT INPUT

Satu bilangan bulat **N**.

FORMAT OUTPUT

YA atau **TIDAK**.

BATASAN

- $1 \leq N \leq 10^9$

Contoh Input 1

64

Contoh Output 1

YA

Contoh Input 2

96

Contoh Output 2

TIDAK

Contoh Input 3

1

Contoh Output 3

YA

Soal 4.14 Tukar dengan XOR[Lihat jawaban →](#)

Tukar nilai dua variabel tanpa variabel bantu, menggunakan operator XOR.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **A** dan **B**.

FORMAT OUTPUT

A dan **B** setelah ditukar, dipisahkan spasi.

BATASAN

- $0 \leq A, B \leq 10^9$

Contoh Input 1

12 30

Contoh Output 1

30 12

Soal 4.15 Bit ke-K[Lihat jawaban →](#)

Cetak nilai bit ke-**K** (dihitung dari kanan, mulai 0) dari bilangan **N**.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **N** dan **K**.

FORMAT OUTPUT

0 atau 1.

BATASAN

- $0 \leq N \leq 10^9$
- $0 \leq K \leq 30$

Contoh Input 1

10 1

Contoh Output 1

1

Contoh Input 2

10 2

Contoh Output 2

0

Soal 4.16 Maksimum dengan Ternary[Lihat jawaban →](#)

Cetak bilangan terbesar dari tiga bilangan menggunakan operator kondisional **? :** saja, tanpa **if**.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat **A**, **B**, dan **C**.

FORMAT OUTPUT

Bilangan terbesar.

BATASAN

- $-10^9 \leq A, B, C \leq 10^9$

Contoh Input 1

4 11 7

Contoh Output 1

11

Soal 4.17 Prioritas Operator[Lihat jawaban →](#)

Hitung tiga ekspresi bilangan bulat berikut untuk nilai A , B , dan C yang diberikan: $A + B * C$, $(A + B) * C$, dan $A - B / C \% 3$.

FORMAT INPUT

Satu baris berisi tiga bilangan bulat A , B , dan C .

FORMAT OUTPUT

Tiga baris berisi hasil ketiga ekspresi.

BATASAN

- $0 \leq A, B \leq 10^4$
- $1 \leq C \leq 10^4$

Contoh Input 1

2 10 4

Contoh Output 1

42
48
0

Soal 4.18 Hari, Jam, dan Menit[Lihat jawaban →](#)

Ubah durasi dalam menit menjadi hari, jam, dan menit.

FORMAT INPUT

Satu bilangan bulat M .

FORMAT OUTPUT

Satu baris dengan format D hari H jam M menit.

BATASAN

- $0 \leq M \leq 10^9$

Contoh Input 1

1500

Contoh Output 1

1 hari 1 jam 0 menit

Contoh Input 2

59

Contoh Output 2

0 hari 0 jam 59 menit

Soal 4.19 Pembulatan Rata-rata[Lihat jawaban →](#)

Hitung rata-rata tiga nilai ujian, lalu bulatkan ke bilangan bulat terdekat (0,5 dibulatkan ke atas).

FORMAT INPUT

Satu baris berisi tiga bilangan bulat A , B , dan C .

FORMAT OUTPUT

Rata-rata yang sudah dibulatkan.

BATASAN

- $0 \leq A, B, C \leq 100$

Contoh Input 1

80 85 91

Contoh Output 1

85

Contoh Input 2

70 71 71

Contoh Output 2

71

Soal 4.20 Kelipatan Terdekat[Lihat jawaban →](#)

Diberikan N dan K . Cetak kelipatan K terbesar yang tidak lebih dari N , dan kelipatan K terkecil yang tidak kurang dari N .

FORMAT INPUT

Satu baris berisi dua bilangan bulat N dan K .

FORMAT OUTPUT

Dua baris berisi kedua kelipatan tersebut.

BATASAN

- $0 \leq N \leq 10^9$
- $1 \leq K \leq 10^4$

Contoh Input 1

47 10

Contoh Output 1

40
50

Contoh Input 2

30 5

Contoh Output 2

30
30

Kontrol Program: Perulangan

Bab ini melatih perulangan for, while, dan do-while, perulangan bersarang (nested loop), sentinel, serta break dan continue. Perhatikan tipe data: hasil perulangan seperti jumlah dan faktorial cepat membesar.

Soal 5.01 Hitung Sampai N

[Lihat jawaban →](#)

Cetak bilangan 1 sampai N , satu bilangan per baris.

FORMAT INPUT

Satu bilangan bulat N .

FORMAT OUTPUT

N baris berisi 1, 2, ..., N .

BATASAN

- $1 \leq N \leq 1.000$

Contoh Input 1

5

Contoh Output 1

1
2
3
4
5

Soal 5.02 Jumlah 1 sampai N

[Lihat jawaban →](#)

Hitung $1 + 2 + 3 + \dots + N$ dengan perulangan.

FORMAT INPUT

Satu bilangan bulat N .

FORMAT OUTPUT

Hasil penjumlahan.

BATASAN

- $1 \leq N \leq 10^6$

Contoh Input 1

10

Contoh Output 1

55

Contoh Input 2

1000000

Contoh Output 2

500000500000

Soal 5.03 Faktorial[Lihat jawaban →](#)

Hitung $N! = 1 \times 2 \times \dots \times N$. Ingat bahwa $0! = 1$.

FORMAT INPUT

Satu bilangan bulat **N**.

FORMAT OUTPUT

Nilai $N!$.

BATASAN

- $0 \leq N \leq 20$

Contoh Input 1

5

Contoh Output 1

120

Contoh Input 2

20

Contoh Output 2

2432902008176640000

Contoh Input 3

0

Contoh Output 3

1

Soal 5.04 Bilangan Ganjil dalam Rentang[Lihat jawaban →](#)

Cetak semua bilangan ganjil dari **A** sampai **B** (inklusif) dalam satu baris, dipisahkan spasi.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **A** dan **B**.

FORMAT OUTPUT

Bilangan-bilangan ganjil dipisahkan satu spasi, tanpa spasi di akhir baris. Dijamin ada minimal satu bilangan ganjil.

BATASAN

- $1 \leq A \leq B \leq 10^5$

Contoh Input 1

11 20

Contoh Output 1

11 13 15 17 19

Contoh Input 2

4 7

Contoh Output 2

5 7

Soal 5.05 Tabel Perkalian[Lihat jawaban →](#)

Cetak tabel perkalian N dari 1 sampai 10.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $1 \leq N \leq 1.000$

Contoh Input 1

7

FORMAT OUTPUT

10 baris dengan format $N \times i = \text{hasil}$.

Contoh Output 1

```
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

Soal 5.06 Segitiga Bintang[Lihat jawaban →](#)

Cetak segitiga siku-siku dari karakter $*$ setinggi N baris. Baris ke- i berisi i buah bintang.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $1 \leq N \leq 100$

Contoh Input 1

4

FORMAT OUTPUT

N baris segitiga bintang.

Contoh Output 1

```
*
**
***
****
```

Soal 5.07 Piramida Bintang[Lihat jawaban →](#)

Cetak piramida bintang setinggi N . Baris ke- i berisi $N - i$ spasi di depan, diikuti $2i - 1$ bintang. Tidak ada spasi di akhir baris.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $1 \leq N \leq 100$

Contoh Input 1

4

FORMAT OUTPUT

N baris piramida.

Contoh Output 1

```
*
 ***
*****
*****
```

Soal 5.08 Jumlah Digit Bilangan Besar[Lihat jawaban →](#)

Hitung jumlah semua digit sebuah bilangan bulat tak negatif yang bisa sangat besar.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $0 \leq N \leq 10^{18}$

Contoh Input 1

98765

Contoh Input 2

0

FORMAT OUTPUT

Jumlah digit N .

Contoh Output 1

35

Contoh Output 2

0

Soal 5.09 Membalik Bilangan[Lihat jawaban →](#)

Balik urutan digit sebuah bilangan bulat tak negatif. Nol di depan hasil tidak dicetak.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Bilangan hasil pembalikan.

BATASAN

- $0 \leq N \leq 10^9$

Contoh Input 1

12345

Contoh Output 1

54321

Contoh Input 2

1200

Contoh Output 2

21

Soal 5.10 Bilangan Prima[Lihat jawaban →](#)

Tentukan apakah `N` adalah bilangan prima, yaitu bilangan lebih dari 1 yang hanya habis dibagi 1 dan dirinya sendiri.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

`PRIMA` atau `BUKAN PRIMA`.

BATASAN

- $1 \leq N \leq 2 \times 10^9$

Contoh Input 1

97

Contoh Output 1

PRIMA

Contoh Input 2

91

Contoh Output 2

BUKAN PRIMA

Contoh Input 3

1

Contoh Output 3

BUKAN PRIMA

Soal 5.11 **FPB dan KPK**[Lihat jawaban →](#)

Hitung Faktor Persekutuan Terbesar (FPB) dan Kelipatan Persekutuan Terkecil (KPK) dari dua bilangan bulat positif.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **A** dan **B**.

FORMAT OUTPUT

Dua baris: FPB lalu KPK.

BATASAN

- $1 \leq A, B \leq 10^9$

Contoh Input 1

12 18

Contoh Output 1

6
36

Contoh Input 2

7 5

Contoh Output 2

1
35

Soal 5.12 **Deret Fibonacci**[Lihat jawaban →](#)

Deret Fibonacci dimulai dengan 1, 1, dan setiap suku berikutnya adalah jumlah dua suku sebelumnya. Cetak **N** suku pertama.

FORMAT INPUT

Satu bilangan bulat **N**.

FORMAT OUTPUT

N suku pertama dalam satu baris, dipisahkan spasi.

BATASAN

- $1 \leq N \leq 90$

Contoh Input 1

10

Contoh Output 1

1 1 2 3 5 8 13 21 34 55

Contoh Input 2

1

Contoh Output 2

1

Soal 5.13 Rata-rata dengan Sentinel[Lihat jawaban →](#)

Baca bilangan-bilangan bulat positif satu per satu sampai bertemu angka 0 (sentinel). Hitung rata-ratanya. Angka 0 tidak ikut dihitung.

FORMAT INPUT

Beberapa bilangan bulat, diakhiri dengan 0.

FORMAT OUTPUT

Rata-rata dengan 2 angka di belakang koma, atau `Tidak ada data` jika bilangan pertama sudah 0.

BATASAN

- Paling banyak 10^5 bilangan sebelum 0.
- $1 \leq \text{setiap bilangan} \leq 10^6$

Contoh Input 1

10 20 25 0

Contoh Output 1

18.33

Contoh Input 2

0

Contoh Output 2

Tidak ada data

Soal 5.14 Desimal ke Biner[Lihat jawaban →](#)

Ubah bilangan bulat tak negatif ke representasi biner tanpa memakai array.

FORMAT INPUT

Satu bilangan bulat `N`.

FORMAT OUTPUT

Representasi biner `N` tanpa nol di depan (untuk $N = 0$, cetak `0`).

BATASAN

- $0 \leq N \leq 10^9$

Contoh Input 1

10

Contoh Output 1

1010

Contoh Input 2

255

Contoh Output 2

11111111

Contoh Input 3

0

Contoh Output 3

0

Soal 5.15 Bilangan Sempurna[Lihat jawaban →](#)

Bilangan sempurna adalah bilangan yang sama dengan jumlah semua pembaginya selain dirinya sendiri, contohnya $6 = 1 + 2 + 3$. Tentukan apakah N bilangan sempurna.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $1 \leq N \leq 10^6$

FORMAT OUTPUT

SEMPURNA atau BUKAN SEMPURNA.

Contoh Input 1

28

Contoh Output 1

SEMPURNA

Contoh Input 2

12

Contoh Output 2

BUKAN SEMPURNA

Contoh Input 3

1

Contoh Output 3

BUKAN SEMPURNA

Soal 5.16 Langkah Collatz[Lihat jawaban →](#)

Mulai dari bilangan N . Jika genap, bagi dua. Jika ganjil, ubah menjadi $3N + 1$. Ulangi sampai mencapai 1. Hitung banyaknya langkah.

FORMAT INPUT

Satu bilangan bulat N .

BATASAN

- $1 \leq N \leq 10^6$

FORMAT OUTPUT

Banyaknya langkah sampai N menjadi 1.

Contoh Input 1

6

Contoh Output 1

8

Contoh Input 2

1

Contoh Output 2

0

Soal 5.17 Jumlah dengan Continue[Lihat jawaban →](#)

Jumlahkan semua bilangan dari 1 sampai N yang tidak habis dibagi K . Gunakan `continue` untuk melewati bilangan yang habis dibagi K .

FORMAT INPUT

Satu baris berisi dua bilangan bulat N dan K .

FORMAT OUTPUT

Hasil penjumlahan.

BATASAN

- $1 \leq N \leq 10^6$
- $1 \leq K \leq 10^6$

Contoh Input 1

10 3

Contoh Output 1

37

Contoh Input 2

5 1

Contoh Output 2

0

Soal 5.18 Tebak Angka[Lihat jawaban →](#)

Dalam permainan tebak angka, angka rahasianya X . Pemain menebak N kali. Tentukan pada tebakan ke berapa pemain pertama kali menebak dengan benar. Gunakan `break` untuk berhenti begitu tebakan benar.

FORMAT INPUT

Baris pertama berisi X dan N . Baris kedua berisi N bilangan tebakan.

FORMAT OUTPUT

Benar pada tebakan ke- i atau `Gagal` jika tidak ada tebakan yang benar.

BATASAN

- $1 \leq X \leq 100$
- $1 \leq N \leq 100$
- $1 \leq \text{setiap tebakan} \leq 100$

Contoh Input 1

42 5
50 25 42 42 10

Contoh Output 1

Benar pada tebakan ke-3

Contoh Input 2

7 3
1 2 3

Contoh Output 2

Gagal

Soal 5.19 Bilangan Armstrong[Lihat jawaban →](#)

Bilangan Armstrong tiga digit adalah bilangan yang sama dengan jumlah pangkat tiga digit-digitnya, contohnya $153 = 1^3 + 5^3 + 3^3$. Cetak semua bilangan Armstrong tiga digit dalam rentang **A** sampai **B**.

FORMAT INPUT

Satu baris berisi dua bilangan bulat **A** dan **B**.

FORMAT OUTPUT

Setiap bilangan Armstrong pada baris tersendiri, atau **Tidak ada** jika tidak ada.

BATASAN

- $100 \leq A \leq B \leq 999$

Contoh Input 1

100 999

Contoh Output 1

153
370
371
407

Contoh Input 2

200 300

Contoh Output 2

Tidak ada

Soal 5.20 Persegi Berongga[Lihat jawaban →](#)

Cetak persegi berukuran **N × N** dari karakter ***** yang bagian dalamnya kosong (diisi spasi). Tidak ada spasi di akhir baris.

FORMAT INPUT

Satu bilangan bulat **N**.

FORMAT OUTPUT

N baris membentuk persegi berongga.

BATASAN

- $2 \leq N \leq 100$

Contoh Input 1

4

Contoh Output 1

* *
* *

Contoh Input 2

2

Contoh Output 2

**
**

Kunci Jawaban dan Pembahasan

Semua solusi ditulis dalam C standar (C99) dan sudah diuji dengan GCC terhadap seluruh contoh di soal. Setiap solusi diikuti penjelasan singkat tentang logikanya.

Bab 1 · Pengenalan Algoritma dan Pemrograman C I

Soal 1.01 Halo, Binusian!

[← Soal](#)

```
#include <stdio.h>

int main() {
    printf("Hello, Binusian!\n");
    return 0;
}
```

Pembahasan. Setiap program C dimulai dari fungsi `main()`. Fungsi `printf` dari `stdio.h` mencetak teks apa adanya, dan `\n` menutup baris. `return 0;` menandakan program selesai tanpa kesalahan.

Soal 1.02 Kartu Identitas

[← Soal](#)

```
#include <stdio.h>

int main() {
    char nama[21];
    int umur;
    scanf("%s %d", nama, &umur);
    printf("Nama: %s\n", nama);
    printf("Umur: %d tahun\n", umur);
    return 0;
}
```

Pembahasan. Nama disimpan dalam variabel teks `char nama[21]` dan dibaca dengan `%s`, sedangkan umur dibaca dengan `%d`. Keduanya lalu dicetak kembali sesuai format. Perhatikan bahwa `scanf` untuk `%s` tidak memakai tanda `&`.

Soal 1.03 Luas dan Keliling Persegi Panjang

← Soal

```
#include <stdio.h>

int main() {
    int p, l;
    scanf("%d %d", &p, &l);
    printf("Luas = %d\n", p * l);
    printf("Keliling = %d\n", 2 * (p + l));
    return 0;
}
```

Pembahasan. Luas = $P \times L$ dan keliling = $2 \times (P + L)$. Nilai terbesar $10.000 \times 10.000 = 100.000.000$ masih muat di tipe `int`.

Soal 1.04 Konversi Detik

← Soal

```
#include <stdio.h>

int main() {
    int s;
    scanf("%d", &s);
    int jam = s / 3600;
    int menit = (s - jam * 3600) / 60;
    int detik = s - jam * 3600 - menit * 60;
    printf("%d jam %d menit %d detik\n", jam, menit, detik);
    return 0;
}
```

Pembahasan. Satu jam = 3600 detik, jadi jam = $S / 3600$ (pembagian bulat). Sisa detik setelah jam diambil dibagi 60 untuk mendapat menit, dan sisanya lagi adalah detik. Ini persis algoritma latihan pseudocode di materi.

Soal 1.05 Rata-rata Tiga Bilangan[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    double rata = (a + b + c) / 3.0;
    printf("%.2f\n", rata);
    return 0;
}
```

Pembahasan. Jumlah ketiga bilangan dibagi `3.0` (bukan `3`) agar hasilnya bilangan pecahan, bukan pembagian bulat. Format `%.2f` mencetak dua angka desimal.

Soal 1.06 Kode ASCII[← Soal](#)

```
#include <stdio.h>

int main() {
    char c;
    scanf("%c", &c);
    printf("%d\n", c);
    return 0;
}
```

Pembahasan. Variabel `char` sebenarnya menyimpan bilangan kecil. Bila dicetak dengan `%d`, yang tampil adalah nilai angkanya, yaitu kode ASCII karakter tersebut.

Soal 1.07 Karakter dari Kode[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%c\n", n);
    return 0;
}
```

Pembahasan. Format `%c` menampilkan sebuah nilai angka sebagai karakter. Jadi cukup membaca angka lalu mencetaknya dengan `%c`.

Soal 1.08 Huruf Berikutnya

← Soal

```
#include <stdio.h>

int main() {
    char c;
    scanf("%c", &c);
    printf("%c\n", c + 1);
    return 0;
}
```

Pembahasan. Huruf `a` sampai `z` memiliki kode ASCII berurutan (97 sampai 122). Menambahkan 1 ke kode sebuah huruf menghasilkan huruf berikutnya.

Soal 1.09 Huruf Kapital

← Soal

```
#include <stdio.h>

int main() {
    char c;
    scanf("%c", &c);
    printf("%c\n", c - 32);
    return 0;
}
```

Pembahasan. Selisih kode ASCII huruf kecil dan kapitalnya selalu 32 (misalnya `a` = 97 dan `A` = 65). Mengurangi 32 dari huruf kecil menghasilkan huruf kapitalnya. Bisa juga ditulis `c - 'a' + 'A'`.

Soal 1.10 Celsius ke Fahrenheit

← Soal

```
#include <stdio.h>

int main() {
    int c;
    scanf("%d", &c);
    double f = c * 9.0 / 5.0 + 32;
    printf("%.2f\n", f);
    return 0;
}
```

Pembahasan. Konstanta `9.0` dan `5.0` memaksa perhitungan dilakukan dalam bilangan pecahan, sehingga hasilnya tidak terpotong.

Soal 1.11 Akar Kuadrat

← Soal

```
#include <stdio.h>
#include <math.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%.3f\n", sqrt(n));
    return 0;
}
```

Pembahasan. Fungsi `sqrt` dari `math.h` mengembalikan `double`. Pada GCC, program yang memakai `math.h` kadang perlu dikompilasi dengan opsi `-lm`.

Soal 1.12 Pangkat

← Soal

```
#include <stdio.h>
#include <math.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%.0f\n", pow(a, b));
    return 0;
}
```

Pembahasan. `pow(a, b)` mengembalikan `double`. Karena hasilnya pasti bilangan bulat (paling besar 10 pangkat 9), format `%.0f` menampilkannya tanpa angka desimal.

Soal 1.13 Volume Balok

← Soal

```
#include <stdio.h>

int main() {
    int p, l, t;
    scanf("%d %d %d", &p, &l, &t);
    printf("Volume = %d\n", p * l * t);
    printf("Luas Permukaan = %d\n", 2 * (p * l + p * t + l * t));
    return 0;
}
```

Pembahasan. $\text{Volume} = P \times L \times T$ dan $\text{luas permukaan} = 2 \times (PL + PT + LT)$. Nilai maksimum 10^9 dan 6×10^6 masih muat dalam `int`.

Soal 1.14 Jarak Dua Titik[← Soal](#)

```
#include <stdio.h>
#include <math.h>

int main() {
    int x1, y1, x2, y2;
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
    double dx = x2 - x1;
    double dy = y2 - y1;
    printf("%.2f\n", sqrt(dx * dx + dy * dy));
    return 0;
}
```

Pembahasan. Rumus jarak Euclid adalah akar dari $(X2 - X1)^2 + (Y2 - Y1)^2$. Selisih koordinat disimpan sebagai `double` supaya perkaliannya tidak meluap.

Soal 1.15 Tukar Nilai[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, temp;
    scanf("%d %d", &a, &b);
    temp = a;
    a = b;
    b = temp;
    printf("%d %d\n", a, b);
    return 0;
}
```

Pembahasan. Nilai `a` disimpan dulu di `temp` agar tidak hilang saat `a` ditimpa oleh `b`. Setelah itu `b` diisi dengan nilai lama `a` dari `temp`.

Soal 1.16 Nilai Mutlak[← Soal](#)

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

int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", abs(n));
    return 0;
}
```

Pembahasan. Fungsi `abs` mengembalikan nilai positif dari bilangan bulat. Fungsi ini termasuk pustaka standar `stdlib.h`.

Soal 1.17 Total Belanja dan Pajak[← Soal](#)

```
#include <stdio.h>

int main() {
    int h, n;
    scanf("%d %d", &h, &n);
    int subtotal = h * n;
    int pajak = subtotal / 10;
    printf("Subtotal: %d\n", subtotal);
    printf("Pajak: %d\n", pajak);
    printf("Total: %d\n", subtotal + pajak);
    return 0;
}
```

Pembahasan. Subtotal = $H \times N$. Karena H kelipatan 10, subtotal juga kelipatan 10 sehingga pajak $10\% = \text{subtotal} / 10$ selalu bulat.

Soal 1.18 Sisi Miring Segitiga Siku-siku[← Soal](#)

```
#include <stdio.h>
#include <math.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    double c = sqrt((double)a * a + (double)b * b);
    printf("%.2f\n", c);
    printf("%.2f\n", a + b + c);
    printf("%.2f\n", a * b / 2.0);
    return 0;
}
```

Pembahasan. Teorema Pythagoras: sisi miring $C = \sqrt{A^2 + B^2}$. Keliling = $A + B + C$ dan luas = $A \times B / 2$. Pembagi `2.0` menjaga hasil luas tetap pecahan.

Soal 1.19 Akar Persamaan Kuadrat[← Soal](#)

```
#include <stdio.h>
#include <math.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    double d = (double)b * b - 4.0 * a * c;
    double x1 = (-b + sqrt(d)) / (2.0 * a);
    double x2 = (-b - sqrt(d)) / (2.0 * a);
    printf("%.2f\n%.2f\n", x1, x2);
    return 0;
}
```

Pembahasan. Diskriminan $D = b^2 - 4ac$. Akar-akarnya $(-b \pm \sqrt{D}) / 2a$. Karena a positif, akar dengan tanda $+$ selalu yang lebih besar, sehingga tidak perlu membandingkan. Ini sesuai contoh flowchart di materi.

```
#include <stdio.h>

int main() {
    int h1, m1, s1, h2, m2, s2;
    scanf("%d %d %d", &h1, &m1, &s1);
    scanf("%d %d %d", &h2, &m2, &s2);
    int mulai = h1 * 3600 + m1 * 60 + s1;
    int selesai = h2 * 3600 + m2 * 60 + s2;
    printf("%d\n", selesai - mulai);
    return 0;
}
```

Pembahasan. Ubah kedua waktu menjadi jumlah detik sejak tengah malam ($\text{jam} \times 3600 + \text{menit} \times 60 + \text{detik}$), lalu kurangkan. Teknik mengubah ke satuan terkecil membuat selisih waktu mudah dihitung.

Bab 2 · Pengenalan Pemrograman C II & Input/Output

Soal 2.01 Nilai Akhir Mata Kuliah

← Soal

```
#include <stdio.h>

int main() {
    int t, m, f;
    scanf("%d %d %d", &t, &m, &f);
    float akhir = 0.2f * t + 0.3f * m + 0.5f * f;
    printf("%.2f\n", akhir);
    return 0;
}
```

Pembahasan. Persentase ditulis sebagai konstanta pecahan. Akhiran `f` (suffix) menandai konstanta bertipe `float`, sehingga tidak ada peringatan konversi dari `double` ke `float`.

Soal 2.02 Pembagian Bulat dan Pecahan

← Soal

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a / b);
    printf("%.3f\n", (double)a / b);
    return 0;
}
```

Pembahasan. `a / b` dengan dua operan `int` menghasilkan `int`. Dengan casting `(double)a`, salah satu operan menjadi pecahan sehingga seluruh pembagian dilakukan sebagai pecahan.

Soal 2.03 Bagian Bulat dan Pecahan

← Soal

```
#include <stdio.h>

int main() {
    double x;
    scanf("%lf", &x);
    int bulat = (int)x;
    double pecahan = x - bulat;
    printf("%d\n", bulat);
    printf("%.2f\n", pecahan);
    return 0;
}
```

Pembahasan. Casting `(int)x` membuang bagian di belakang koma. Bagian pecahan didapat dengan mengurangkan bagian bulat dari `x`. Perhatikan `double` dibaca dengan `%lf`.

Soal 2.04 Lingkaran dengan Konstanta

← Soal

```
#include <stdio.h>
#define PI 3.14159

int main() {
    int r;
    scanf("%d", &r);
    printf("%.3f\n", PI * r * r);
    printf("%.3f\n", 2 * PI * r);
    return 0;
}
```

Pembahasan. `#define PI 3.14159` adalah konstanta simbolik: preprocessor mengganti setiap `PI` dengan nilainya sebelum kompilasi, dan nilainya tidak bisa diubah saat program berjalan.

Soal 2.05 Ukuran Tipe Data

← Soal

```
#include <stdio.h>

int main() {
    printf("char: %d\n", (int)sizeof(char));
    printf("int: %d\n", (int)sizeof(int));
    printf("float: %d\n", (int)sizeof(float));
    printf("double: %d\n", (int)sizeof(double));
    return 0;
}
```

Pembahasan. `sizeof` mengembalikan ukuran tipe dalam byte. Hasilnya bertipe `size_t`, jadi di-casting ke `int` agar aman dicetak dengan `%d` di semua kompiler. Ukurannya bisa berbeda di kompiler lama, misalnya `int` 2 byte di Turbo C.

Soal 2.06 Kolom Rata Kanan

← Soal

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    printf("%8d\n%8d\n%8d\n", a, b, c);
    return 0;
}
```

Pembahasan. Angka di antara `%` dan `d` adalah lebar kolom. `%8d` menyediakan 8 kolom dan mengisi sisanya dengan spasi di kiri, sehingga angka rata kanan.

Soal 2.07 Label Harga

← Soal

```
#include <stdio.h>

int main() {
    char nama[11];
    double harga;
    scanf("%s %lf", nama, &harga);
    printf("|%-10s|%10.2f|\n", nama, harga);
    return 0;
}
```

Pembahasan. Flag `-` membuat isi rata kiri (`%-10s`). Pada `%10.2f`, angka 10 adalah lebar kolom dan `.2` adalah presisi, yaitu jumlah angka di belakang koma.

Soal 2.08 Sapaan Nama Lengkap[← Soal](#)

```
#include <stdio.h>

int main() {
    char nama[101];
    scanf("%[^\\n]", nama);
    printf("Halo, %s!\\n", nama);
    return 0;
}
```

Pembahasan. `%s` berhenti membaca di spasi pertama, jadi hanya mendapat `Budi`. Format `%[^]` membaca semua karakter sampai ketemu baris baru, sehingga nama lengkap terbaca utuh.

Soal 2.09 Format Tanggal[← Soal](#)

```
#include <stdio.h>

int main() {
    int d, m, y;
    scanf("%d/%d/%d", &d, &m, &y);
    printf("%04d-%02d-%02d\\n", y, m, d);
    return 0;
}
```

Pembahasan. Karakter `/` di dalam format `scanf` harus cocok dengan input, sehingga ketiga angka terbaca terpisah. Flag `0` pada `%02d` mengisi kolom kosong dengan nol, sehingga 1 menjadi `01`.

Soal 2.10 Menit Sejak Tengah Malam[← Soal](#)

```
#include <stdio.h>

int main() {
    int jam, menit;
    scanf("%d:%d", &jam, &menit);
    printf("%d\\n", jam * 60 + menit);
    return 0;
}
```

Pembahasan. Tanda `:` di format `scanf` memisahkan jam dan menit. Hasilnya $\text{jam} \times 60 + \text{menit}$.

Soal 2.11 Oktal dan Heksadesimal

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%o\n%x\n%X\n", n, n, n);
    return 0;
}
```

Pembahasan. `printf` bisa langsung mencetak basis lain: `%o` untuk oktal, `%x` untuk heksadesimal huruf kecil, dan `%X` untuk huruf besar.

Soal 2.12 Membaca Heksadesimal

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%x", &n);
    printf("%d\n", n);
    return 0;
}
```

Pembahasan. Format `%x` pada `scanf` membaca input sebagai bilangan heksadesimal. Nilainya disimpan sebagai `int` biasa, lalu dicetak dengan `%d`.

Soal 2.13 Dua Karakter Terbalik

← Soal

```
#include <stdio.h>

int main() {
    char c1 = getchar();
    char c2 = getchar();
    putchar(c2);
    putchar(c1);
    putchar('\n');
    return 0;
}
```

Pembahasan. `getchar()` mengambil satu karakter dari input setiap kali dipanggil. `putchar` mencetak satu karakter.

Soal 2.14 Singkatan Tiga Huruf[← Soal](#)

```
#include <stdio.h>

int main() {
    char kode[4];
    scanf("%3s", kode);
    printf("%s\n", kode);
    return 0;
}
```

Pembahasan. Angka pada `%3s` membatasi `scanf` agar membaca paling banyak 3 karakter. Array berukuran 4 karena perlu satu tempat lagi untuk karakter penutup `\0`.

Soal 2.15 Perkalian Besar[← Soal](#)

```
#include <stdio.h>

int main() {
    long long a, b;
    scanf("%lld %lld", &a, &b);
    printf("%lld\n", a * b);
    return 0;
}
```

Pembahasan. `int` hanya menampung sekitar $\pm 2,1 \times 10^9$, sedangkan hasil kali bisa mencapai 10^{18} . Modifier `long long` menampung sampai sekitar $9,2 \times 10^{18}$ dan dicetak dengan `%lld`.

Soal 2.16 Meluap di Unsigned Char[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    unsigned char c = (unsigned char)n;
    printf("%d\n", c);
    return 0;
}
```

Pembahasan. `unsigned char` berukuran 8 bit, jadi hanya 8 bit terbawah yang tersimpan. Hasilnya sama dengan sisa bagi `N` oleh 256: $300 - 256 = 44$.

Soal 2.17 Struk Belanja[← Soal](#)

```
#include <stdio.h>

int main() {
    char n1[11], n2[11];
    int q1, h1, q2, h2;
    scanf("%s %d %d", n1, &q1, &h1);
    scanf("%s %d %d", n2, &q2, &h2);
    long long s1 = (long long)q1 * h1;
    long long s2 = (long long)q2 * h2;
    printf("%-10s%5d%12lld\n", n1, q1, s1);
    printf("%-10s%5d%12lld\n", n2, q2, s2);
    printf("%-15s%12lld\n", "TOTAL", s1 + s2);
    return 0;
}
```

Pembahasan. Lebar kolom dipilih agar semua baris sama panjang (27 karakter): $10 + 5 + 12 = 15 + 12$. Subtotal bisa mencapai 10^9 dan totalnya 2×10^9 , jadi dipakai `long long`.

Soal 2.18 Kalimat dalam Tanda Kutip[← Soal](#)

```
#include <stdio.h>

int main() {
    char s[201];
    scanf("%[^\n]", s);
    printf("\"%s\"\n", s);
    return 0;
}
```

Pembahasan. Tanda kutip ganda di dalam string C harus ditulis dengan escape sequence `\"`. Tanpa backslash, kompiler mengira string sudah berakhir.

Soal 2.19 Label Diskon

← Soal

```
#include <stdio.h>

int main() {
    int h, d;
    scanf("%d %d", &h, &d);
    int potongan = h / 100 * d;
    printf("Diskon %d%%: Rp %d\n", d, potongan);
    printf("Bayar: Rp %d\n", h - potongan);
    return 0;
}
```

Pembahasan. Untuk mencetak tanda persen di `printf`, tulis `%%`. Karena `H` kelipatan 100, `H / 100 * D` selalu tepat. Membagi dulu sebelum mengali juga mencegah `int` meluap.

Soal 2.20 Angka dari Karakter

← Soal

```
#include <stdio.h>

int main() {
    char c = getchar();
    int nilai = c - '0';
    printf("%d %d\n", nilai, nilai * nilai);
    return 0;
}
```

Pembahasan. Karakter `'0'` sampai `'9'` memiliki kode ASCII 48 sampai 57. Mengurangi `'0'` dari karakter digit menghasilkan nilai angkanya, misalnya `'7' - '0' = 55 - 48 = 7`.

Bab 3 • Kontrol Program: Seleksi

Soal 3.01 Ganjil atau Genap

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n % 2 == 0)
        printf("GENAP\n");
    else
        printf("GANJIL\n");
    return 0;
}
```

Pembahasan. Bilangan genap habis dibagi 2, artinya `n % 2 == 0`. Mengecek `n % 2 == 1` untuk ganjil berbahaya, karena untuk bilangan negatif ganjil hasil `%` di C adalah `-1`, bukan `1`.

Soal 3.02 Positif, Negatif, atau Nol

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n > 0)
        printf("POSITIF\n");
    else if (n < 0)
        printf("NEGATIF\n");
    else
        printf("NOL\n");
    return 0;
}
```

Pembahasan. Ada tiga kemungkinan yang saling lepas, jadi cocok memakai rantai `if - else if - else`. Kasus terakhir (`else`) otomatis berarti `n == 0`.

Soal 3.03 Terbesar dari Tiga[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    int maks = a;
    if (b > maks) maks = b;
    if (c > maks) maks = c;
    printf("%d\n", maks);
    return 0;
}
```

Pembahasan. Anggap **A** yang terbesar, lalu bandingkan dengan **B** dan **C** satu per satu dan perbarui jika ada yang lebih besar. Cara ini juga benar ketika ada nilai yang sama.

Soal 3.04 Tahun Kabisat[← Soal](#)

```
#include <stdio.h>

int main() {
    int y;
    scanf("%d", &y);
    if (y % 400 == 0)
        printf("KABISAT\n");
    else if (y % 100 == 0)
        printf("BUKAN KABISAT\n");
    else if (y % 4 == 0)
        printf("KABISAT\n");
    else
        printf("BUKAN KABISAT\n");
    return 0;
}
```

Pembahasan. Urutan pengecekan penting: syarat paling khusus (habis dibagi 400) dicek lebih dulu, lalu 100, lalu 4. Jika urutannya dibalik, tahun 1900 akan salah dianggap kabisat karena habis dibagi 4.

Soal 3.05 Nilai Huruf[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n >= 85) printf("A\n");
    else if (n >= 70) printf("B\n");
    else if (n >= 60) printf("C\n");
    else if (n >= 50) printf("D\n");
    else printf("E\n");
    return 0;
}
```

Pembahasan. Karena kondisi dicek dari batas tertinggi, setiap `else if` otomatis berarti nilai di bawah batas sebelumnya. Jadi cukup mengecek batas bawah tiap rentang.

Soal 3.06 Predikat Kelulusan[← Soal](#)

```
#include <stdio.h>

int main() {
    double ipk;
    scanf("%lf", &ipk);
    if (ipk >= 3.5) printf("Cumlaude\n");
    else if (ipk >= 3.0) printf("Sangat Memuaskan\n");
    else if (ipk >= 2.5) printf("Memuaskan\n");
    else if (ipk >= 2.0) printf("Cukup\n");
    else printf("Kurang\n");
    return 0;
}
```

Pembahasan. Polanya sama dengan soal nilai huruf, tetapi dengan bilangan real. Memakai batas bawah (`>= 3.5`, dan seterusnya) lebih aman daripada menulis batas atas seperti `<= 3.49`, yang bisa melewati nilai 3,495.

Soal 3.07 Kalkulator Sederhana[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b;
    char op;
    scanf("%d %c %d", &a, &op, &b);
    switch (op) {
        case '+': printf("%d\n", a + b); break;
        case '-': printf("%d\n", a - b); break;
        case '*': printf("%d\n", a * b); break;
        case '/':
            if (b == 0) printf("Error\n");
            else printf("%d\n", a / b);
            break;
    }
    return 0;
}
```

Pembahasan. `switch` mencocokkan karakter operator dengan setiap `case`. Jangan lupa `break` agar eksekusi tidak berlanjut ke `case` berikutnya. Pembagian dengan nol harus dicegah karena menyebabkan run-time error.

Soal 3.08 Nama Hari[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    switch (n) {
        case 1: printf("Senin\n"); break;
        case 2: printf("Selasa\n"); break;
        case 3: printf("Rabu\n"); break;
        case 4: printf("Kamis\n"); break;
        case 5: printf("Jumat\n"); break;
        case 6: printf("Sabtu\n"); break;
        case 7: printf("Minggu\n"); break;
        default: printf("Tidak valid\n");
    }
    return 0;
}
```

Pembahasan. `switch-case` sangat cocok untuk memetakan nilai tertentu ke keluaran tertentu. `default` menangani semua nilai yang tidak cocok dengan `case` mana pun.

Soal 3.09 Jenis Segitiga[← Soal](#)

```
#include <stdio.h>

int main() {
    long long a, b, c;
    scanf("%lld %lld %lld", &a, &b, &c);
    if (a + b <= c || a + c <= b || b + c <= a)
        printf("Bukan Segitiga\n");
    else if (a == b && b == c)
        printf("Sama Sisi\n");
    else if (a == b || b == c || a == c)
        printf("Sama Kaki\n");
    else
        printf("Sembarang\n");
    return 0;
}
```

Pembahasan. Tiga sisi membentuk segitiga jika jumlah dua sisi mana pun lebih besar dari sisi ketiga. Setelah itu, cek sama sisi lebih dulu, karena segitiga sama sisi juga memenuhi syarat sama kaki. `long long` dipakai agar penjumlahan dua sisi hingga 2×10^9 tidak meluap.

Soal 3.10 Jenis Akar Persamaan Kuadrat[← Soal](#)

```
#include <stdio.h>
#include <math.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    int d = b * b - 4 * a * c;
    if (d > 0) {
        double x1 = (-b + sqrt(d)) / (2.0 * a);
        double x2 = (-b - sqrt(d)) / (2.0 * a);
        printf("%.2f\n%.2f\n", x1, x2);
    } else if (d == 0) {
        double x = -b / (2.0 * a);
        if (x == 0) x = 0; /* hindari tampilan -0.00 */
        printf("%.2f\n", x);
    } else {
        printf("Akar imajiner\n");
    }
    return 0;
}
```

Pembahasan. Nilai diskriminan $D = b^2 - 4ac$ menentukan jumlah akar real: dua, satu (kembar), atau tidak ada. Untuk kasus $b = 0$ dan $D = 0$, hasil `-0 / 2a` bisa tercetak `-0.00`, jadi nilai nol dipaksa menjadi 0 positif.

Soal 3.11 Habis Dibagi 3 atau 5

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    if (n % 3 == 0 && n % 5 == 0) printf("FizzBuzz\n");
    else if (n % 3 == 0) printf("Fizz\n");
    else if (n % 5 == 0) printf("Buzz\n");
    else printf("%d\n", n);
    return 0;
}
```

Pembahasan. Kondisi gabungan (habis dibagi 3 dan 5) harus dicek paling awal. Jika `n % 3 == 0` dicek dulu, angka 15 akan berhenti di `Fizz`. Kesalahan urutan seperti ini dibahas di materi sebagai contoh if yang salah.

Soal 3.12 Tarif Parkir

← Soal

```
#include <stdio.h>

int main() {
    int j, tarif;
    scanf("%d", &j);
    if (j <= 2)
        tarif = 5000;
    else
        tarif = 5000 + (j - 2) * 3000;
    if (tarif > 30000)
        tarif = 30000;
    printf("%d\n", tarif);
    return 0;
}
```

Pembahasan. Pisahkan dua kasus: paling lama 2 jam (tarif tetap) dan lebih dari 2 jam (tarif tetap ditambah tarif per jam tambahan). Batas maksimum diterapkan dengan `if` terpisah setelahnya.

Soal 3.13 Kuadran Titik[← Soal](#)

```
#include <stdio.h>

int main() {
    int x, y;
    scanf("%d %d", &x, &y);
    if (x == 0 && y == 0) printf("Titik Asal\n");
    else if (y == 0) printf("Sumbu X\n");
    else if (x == 0) printf("Sumbu Y\n");
    else if (x > 0) {
        if (y > 0) printf("Kuadran I\n");
        else printf("Kuadran IV\n");
    } else {
        if (y > 0) printf("Kuadran II\n");
        else printf("Kuadran III\n");
    }
    return 0;
}
```

Pembahasan. Kasus pada sumbu dicek lebih dulu. Setelah itu dipakai nested if: tanda X menentukan sisi kanan atau kiri, lalu tanda Y menentukan atas atau bawah.

Soal 3.14 Diskon Bertingkat[← Soal](#)

```
#include <stdio.h>

int main() {
    long long t, bayar;
    scanf("%lld", &t);
    if (t >= 500000)
        bayar = t * 80 / 100;
    else if (t >= 200000)
        bayar = t * 90 / 100;
    else
        bayar = t;
    printf("%lld\n", bayar);
    return 0;
}
```

Pembahasan. Membayar setelah diskon 20% sama dengan membayar 80% harga. `t * 80 / 100` dengan bilangan bulat otomatis membulatkan ke bawah. `long long` dipakai karena `t * 90` bisa melebihi batas `int`.

Soal 3.15 Vokal atau Konsonan

[← Soal](#)

```
#include <stdio.h>

int main() {
    char c;
    scanf("%c", &c);
    switch (c) {
        case 'a': case 'i': case 'u': case 'e': case 'o':
        case 'A': case 'I': case 'U': case 'E': case 'O':
            printf("Vokal\n");
            break;
        default:
            if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))
                printf("Konsonan\n");
            else
                printf("Bukan Huruf\n");
    }
    return 0;
}
```

Pembahasan. Beberapa `case` tanpa `break` di antaranya akan menjalankan perintah yang sama (fall-through), sehingga kesepuluh huruf vokal ditangani bersama. Selain vokal, karakter dicek apakah berada di rentang huruf.

Soal 3.16 Jumlah Hari dalam Sebulan

[← Soal](#)

```
#include <stdio.h>

int main() {
    int m, y, hari;
    scanf("%d %d", &m, &y);
    switch (m) {
        case 2:
            if ((y % 4 == 0 && y % 100 != 0) || y % 400 == 0) hari = 29;
            else hari = 28;
            break;
        case 4: case 6: case 9: case 11:
            hari = 30;
            break;
        default:
            hari = 31;
    }
    printf("%d\n", hari);
    return 0;
}
```

Pembahasan. Bulan dengan 30 hari dikelompokkan dalam satu set `case`. Februari dicek dengan syarat kabisat. Semua bulan lain memiliki 31 hari dan ditangani oleh `default`.

Soal 3.17 Indeks Massa Tubuh[← Soal](#)

```
#include <stdio.h>

int main() {
    int b, t;
    scanf("%d %d", &b, &t);
    double m = t / 100.0;
    double imt = b / (m * m);
    printf("%.1f ", imt);
    if (imt < 18.5) printf("Kurus\n");
    else if (imt < 25) printf("Normal\n");
    else if (imt < 30) printf("Gemuk\n");
    else printf("Obesitas\n");
    return 0;
}
```

Pembahasan. Tinggi diubah dari cm ke meter dengan membagi `100.0`. Kategori ditentukan dengan rantai `else if` dari batas terkecil, sehingga cukup memakai satu pembanding di setiap langkah.

Soal 3.18 Urutkan Tiga Bilangan[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, c, t;
    scanf("%d %d %d", &a, &b, &c);
    if (a > b) { t = a; a = b; b = t; }
    if (b > c) { t = b; b = c; c = t; }
    if (a > b) { t = a; a = b; b = t; }
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

Pembahasan. Tiga kali tukar bersyarat sudah cukup. Dua langkah pertama memastikan yang terbesar berada di `c`. Langkah ketiga merapikan urutan `a` dan `b`.

Soal 3.19 Tagihan Listrik Bertingkat[← Soal](#)

```
#include <stdio.h>

int main() {
    long long k, total;
    scanf("%lld", &k);
    if (k <= 100)
        total = k * 1000;
    else if (k <= 200)
        total = 100 * 1000 + (k - 100) * 1500;
    else
        total = 100 * 1000 + 100 * 1500 + (k - 200) * 2000;
    printf("%lld\n", total);
    return 0;
}
```

Pembahasan. Setiap tingkat hanya mengenakan tarifnya pada bagian pemakaian di dalam rentangnya. Tingkat yang sudah penuh dihitung sebagai konstanta: 100×1.000 dan 100×1.500 .

Soal 3.20 Format Jam 12[← Soal](#)

```
#include <stdio.h>

int main() {
    int h, m;
    scanf("%d:%d", &h, &m);
    int h12 = h % 12;
    if (h12 == 0) h12 = 12;
    printf("%02d:%02d %s\n", h12, m, (h < 12) ? "AM" : "PM");
    return 0;
}
```

Pembahasan. Operator ternary memilih `AM` atau `PM` dari nilai jam. Jam dikonversi dengan `h % 12`. Hasil 0 (yaitu pukul 00 dan 12) diganti menjadi 12.

Bab 4 · Operator, Operan, dan Aritmatika

Soal 4.01 Lima Operasi Dasar

← Soal

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n%d\n%d\n%d\n%d\n", a + b, a - b, a * b, a / b, a % b);
    return 0;
}
```

Pembahasan. Operator `/` pada dua `int` menghasilkan hasil bagi bulat, sedangkan `%` (modulo) menghasilkan sisa baginya. Selalu berlaku $A = (A / B) \times B + A \% B$.

Soal 4.02 Jumlah Digit

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int ratusan = n / 100;
    int puluhan = n / 10 % 10;
    int satuan = n % 10;
    printf("%d\n", ratusan + puluhan + satuan);
    return 0;
}
```

Pembahasan. `n % 10` mengambil digit terakhir, `n / 10 % 10` mengambil digit puluhan, dan `n / 100` mengambil digit ratusan. Operator `/` dan `%` memiliki prioritas sama dan dievaluasi dari kiri ke kanan.

Soal 4.03 Balik Tiga Digit[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int a = n / 100, b = n / 10 % 10, c = n % 10;
    printf("%d\n", c * 100 + b * 10 + a);
    return 0;
}
```

Pembahasan. Pisahkan ketiga digit dengan `/` dan `%`, lalu susun kembali dengan nilai tempat yang dibalik. Jika digit satuan bernilai 0, hasilnya otomatis kurang dari 100, sehingga nol di depan tidak tercetak.

Soal 4.04 Pecahan Uang[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("50000: %d\n", n / 50000); n %= 50000;
    printf("20000: %d\n", n / 20000); n %= 20000;
    printf("10000: %d\n", n / 10000); n %= 10000;
    printf("5000: %d\n", n / 5000); n %= 5000;
    printf("2000: %d\n", n / 2000); n %= 2000;
    printf("1000: %d\n", n / 1000);
    return 0;
}
```

Pembahasan. Untuk setiap pecahan, jumlah lembar = sisa uang / pecahan, lalu sisa uang diperbarui dengan operator gabungan `n %= pecahan`. Untuk pecahan uang rupiah ini, mengambil pecahan terbesar dulu selalu menghasilkan jumlah lembar paling sedikit.

Soal 4.05 Increment dan Decrement

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", ++n);
    printf("%d\n", n++);
    printf("%d\n", n);
    return 0;
}
```

Pembahasan. Pre-increment `++n` menambah dulu lalu memakai nilainya, jadi tercetak $N + 1$. Post-increment `n++` memakai nilai lama dulu ($N + 1$) baru menambah, sehingga baris terakhir mencetak $N + 2$.

Soal 4.06 Operator Gabungan

← Soal

```
#include <stdio.h>

int main() {
    int x, y;
    scanf("%d %d", &x, &y);
    x *= y + 1; printf("%d\n", x);
    x -= 3;     printf("%d\n", x);
    x /= 2;     printf("%d\n", x);
    x %= 7;     printf("%d\n", x);
    return 0;
}
```

Pembahasan. `x *= y + 1` berarti `x = x * (y + 1)`, bukan `x = x * y + 1`, karena seluruh ruas kanan dihitung lebih dulu. Ini persis contoh soal di materi.

Soal 4.07 Operator Relasional

← Soal

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n%d\n%d\n", a == b, a != b, a < b);
    printf("%d\n%d\n%d\n", a > b, a <= b, a >= b);
    return 0;
}
```

Pembahasan. Dalam C, hasil operator relasional adalah bilangan `int` : 1 untuk benar dan 0 untuk salah. Hasil itu bisa langsung dicetak dengan `%d`.

Soal 4.08 Operator Logika

← Soal

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a && b);
    printf("%d\n", a || b);
    printf("%d\n", !a);
    printf("%d\n", !(a && b));
    return 0;
}
```

Pembahasan. Operator logika memperlakukan setiap nilai bukan nol (termasuk bilangan negatif) sebagai benar, dan hasilnya selalu 0 atau 1. Jangan tertukar dengan operator bitwise `&` dan `|`, yang bekerja per bit.

Soal 4.09 AND, OR, XOR

← Soal

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n%d\n%d\n", a & b, a | b, a ^ b);
    return 0;
}
```

Pembahasan. Operator bitwise membandingkan bit per bit. 24 = 011000 dan 35 = 100011 tidak punya bit 1 yang sama, jadi AND = 0 dan OR = XOR = 111011 = 59, persis contoh di materi.

Soal 4.10 Komplemen Satu[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%d\n", ~n);
    return 0;
}
```

Pembahasan. ~ membalik setiap bit (0 menjadi 1 dan sebaliknya). Pada representasi komplemen dua yang dipakai komputer modern, hasilnya selalu sama dengan $-N - 1$.

Soal 4.11 Geser Bit[← Soal](#)

```
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d %d", &n, &k);
    printf("%d\n%d\n", n << k, n >> k);
    return 0;
}
```

Pembahasan. Menggeser K bit ke kiri sama dengan mengalikan 2^K , sedangkan menggeser ke kanan sama dengan membagi 2^K (dibulatkan ke bawah). Batasan menjamin $N \times 2^{10}$ tidak melebihi kapasitas `int`.

Soal 4.12 Ganjil Genap dengan Bitwise[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%s\n", (n & 1) ? "GANJIL" : "GENAP");
    return 0;
}
```

Pembahasan. Bit paling kanan sebuah bilangan biner bernilai 1 tepat ketika bilangan itu ganjil, dan `n & 1` mengambil bit tersebut. Operator kondisional `? :` memilih teks yang dicetak.

Soal 4.13 Pangkat Dua[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%s\n", (n & (n - 1)) == 0 ? "YA" : "TIDAK");
    return 0;
}
```

Pembahasan. Pangkat dua hanya memiliki satu bit 1, misalnya $64 = 1000000$. Mengurangi 1 membalik bit itu dan semua bit di kanannya ($63 = 0111111$), sehingga `n & (n - 1)` bernilai 0. Tanda kurung wajib dipakai karena `==` berprioritas lebih tinggi daripada `&`.

Soal 4.14 Tukar dengan XOR[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    a ^= b;
    b ^= a;
    a ^= b;
    printf("%d %d\n", a, b);
    return 0;
}
```

Pembahasan. XOR memiliki sifat $x \oplus x = 0$ dan $x \oplus 0 = x$. Setelah `a ^= b`, a berisi $a \oplus b$. Kemudian `b ^= a` menghasilkan $b \oplus a \oplus b = a$ (nilai a lama), lalu `a ^= b` menghasilkan $a \oplus b \oplus a = b$.

Soal 4.15 Bit ke-K[← Soal](#)

```
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d %d", &n, &k);
    printf("%d\n", (n >> k) & 1);
    return 0;
}
```

Pembahasan. 10 dalam biner adalah 1010. Menggeser K bit ke kanan membawa bit ke-K ke posisi paling kanan, lalu `& 1` membuang bit lainnya.

Soal 4.16 Maksimum dengan Ternary

← Soal

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    int ab = (a > b) ? a : b;
    int maks = (ab > c) ? ab : c;
    printf("%d\n", maks);
    return 0;
}
```

Pembahasan. Operator ternary `kondisi ? x : y` menghasilkan `x` jika kondisi benar dan `y` jika salah. Cari dulu yang terbesar dari A dan B, lalu bandingkan dengan C.

Soal 4.17 Prioritas Operator

← Soal

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    printf("%d\n", a + b * c);
    printf("%d\n", (a + b) * c);
    printf("%d\n", a - b / c % 3);
    return 0;
}
```

Pembahasan. `*`, `/`, dan `%` lebih diprioritaskan daripada `+` dan `-`, dan operator yang setingkat dievaluasi dari kiri ke kanan. Jadi `A - B / C % 3` dihitung sebagai `A - ((B / C) % 3)`: $10 / 4 = 2$, $2 \% 3 = 2$, $2 - 2 = 0$.

Soal 4.18 Hari, Jam, dan Menit[← Soal](#)

```
#include <stdio.h>

int main() {
    int m;
    scanf("%d", &m);
    int hari = m / 1440;
    int jam = m % 1440 / 60;
    int menit = m % 60;
    printf("%d hari %d jam %d menit\n", hari, jam, menit);
    return 0;
}
```

Pembahasan. Satu hari = 1.440 menit. Sisa menit setelah hari diambil (`m % 1440`) dibagi 60 untuk mendapat jam. Menit terakhir adalah `m % 60`, karena 1.440 adalah kelipatan 60.

Soal 4.19 Pembulatan Rata-rata[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    double rata = (a + b + c) / 3.0;
    printf("%d\n", (int)(rata + 0.5));
    return 0;
}
```

Pembahasan. Casting ke `int` selalu membuang bagian pecahan. Menambahkan 0,5 lebih dulu membuat nilai dengan pecahan 0,5 atau lebih melewati bilangan bulat berikutnya, sehingga hasil casting menjadi pembulatan ke terdekat untuk bilangan positif.

```
#include <stdio.h>

int main() {
    long long n, k;
    scanf("%lld %lld", &n, &k);
    printf("%lld\n", n / k * k);
    printf("%lld\n", (n + k - 1) / k * k);
    return 0;
}
```

Pembahasan. Pembagian bulat n / k lalu dikali k membulatkan ke bawah ke kelipatan K. Menambah $k - 1$ sebelum membagi membuat hasil bagi naik satu kecuali n sudah kelipatan K, sehingga didapat pembulatan ke atas.

Bab 5 • Kontrol Program: Perulangan

Soal 5.01 Hitung Sampai N

← Soal

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
        printf("%d\n", i);
    return 0;
}
```

Pembahasan. Perulangan `for` terdiri dari inisialisasi (`i = 1`), kondisi (`i <= n`), dan perubahan (`i++`). Badan perulangan berjalan selama kondisinya benar.

Soal 5.02 Jumlah 1 sampai N

← Soal

```
#include <stdio.h>

int main() {
    int n;
    long long total = 0;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
        total += i;
    printf("%lld\n", total);
    return 0;
}
```

Pembahasan. Variabel penampung `total` dimulai dari 0 lalu ditambah setiap nilai `i`. Untuk $N = 10^6$ hasilnya sekitar 5×10^{11} , melebihi `int`, sehingga dipakai `long long`.

Soal 5.03 Faktorial[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    long long f = 1;
    scanf("%d", &n);
    for (int i = 2; i <= n; i++)
        f *= i;
    printf("%lld\n", f);
    return 0;
}
```

Pembahasan. Hasil kali dimulai dari 1, bukan 0. Untuk $N = 0$ atau 1, perulangan tidak berjalan sama sekali sehingga hasilnya tetap 1. $20! \approx 2,4 \times 10^{18}$ masih muat di `long long`.

Soal 5.04 Bilangan Ganjil dalam Rentang[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    int mulai = (a % 2 == 1) ? a : a + 1;
    printf("%d", mulai);
    for (int i = mulai + 2; i <= b; i += 2)
        printf(" %d", i);
    printf("\n");
    return 0;
}
```

Pembahasan. Tentukan bilangan ganjil pertama, lalu loncat 2 setiap langkah. Bilangan pertama dicetak tanpa spasi, dan bilangan berikutnya dicetak dengan spasi di depannya, sehingga tidak ada spasi berlebih di akhir baris.

Soal 5.05 Tabel Perkalian[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= 10; i++)
        printf("%d x %d = %d\n", n, i, n * i);
    return 0;
}
```

Pembahasan. Jumlah perulangan sudah pasti (10 kali), jadi `for` adalah pilihan yang paling alami.

Soal 5.06 Segitiga Bintang[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

Pembahasan. Nested loop: perulangan luar mengatur baris, sedangkan perulangan dalam mencetak bintang sebanyak nomor barisnya.

Soal 5.07 Piramida Bintang[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++) printf(" ");
        for (int j = 1; j <= 2 * i - 1; j++) printf("*");
        printf("\n");
    }
    return 0;
}
```

Pembahasan. Setiap baris terdiri dari dua bagian yang dicetak oleh dua perulangan dalam: spasi untuk menggeser ke tengah, lalu bintang. Jumlah keduanya ditentukan dari nomor baris `i`.

Soal 5.08 Jumlah Digit Bilangan Besar[← Soal](#)

```
#include <stdio.h>

int main() {
    long long n;
    int total = 0;
    scanf("%lld", &n);
    while (n > 0) {
        total += n % 10;
        n /= 10;
    }
    printf("%d\n", total);
    return 0;
}
```

Pembahasan. Banyaknya digit belum diketahui, jadi dipakai `while`. Setiap putaran mengambil digit terakhir (`n % 10`) lalu membuangnya (`n /= 10`), sampai `n` habis menjadi 0.

Soal 5.09 Membalik Bilangan[← Soal](#)

```
#include <stdio.h>

int main() {
    long long n, balik = 0;
    scanf("%lld", &n);
    while (n > 0) {
        balik = balik * 10 + n % 10;
        n /= 10;
    }
    printf("%lld\n", balik);
    return 0;
}
```

Pembahasan. Setiap digit terakhir `n` dipindahkan ke ujung kanan `balik` dengan cara `balik * 10 + digit`. Hasilnya bisa melebihi `int` (misalnya 1.999.999.999 dibalik menjadi 9.999.999.991), sehingga dipakai `long long`.

Soal 5.10 Bilangan Prima[← Soal](#)

```
#include <stdio.h>

int main() {
    long long n;
    scanf("%lld", &n);
    int prima = (n >= 2);
    for (long long i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            prima = 0;
            break;
        }
    }
    printf("%s\n", prima ? "PRIMA" : "BUKAN PRIMA");
    return 0;
}
```

Pembahasan. Jika `N` punya pembagi selain 1 dan dirinya, salah satunya pasti tidak lebih besar dari $\sqrt{N}$. Jadi cukup mencoba `i` sampai `i * i <= n` (sekitar 45.000 kali), dan `break` begitu pembagi pertama ditemukan.

Soal 5.11 **FPB dan KPK**[← Soal](#)

```
#include <stdio.h>

int main() {
    long long a, b;
    scanf("%lld %lld", &a, &b);
    long long x = a, y = b;
    while (y != 0) {
        long long sisa = x % y;
        x = y;
        y = sisa;
    }
    long long fpb = x;
    printf("%lld\n%lld\n", fpb, a / fpb * b);
    return 0;
}
```

Pembahasan. Algoritma Euclid: $\text{FPB}(x, y) = \text{FPB}(y, x \bmod y)$, diulang sampai $y = 0$. Dari sifat $\text{FPB} \times \text{KPK} = A \times B$, $\text{KPK} = A / \text{FPB} \times B$. Pembagian dilakukan dulu agar hasil sementara tidak terlalu besar.

Soal 5.12 **Deret Fibonacci**[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    long long a = 1, b = 1;
    for (int i = 1; i <= n; i++) {
        if (i > 1) printf(" ");
        printf("%lld", a);
        long long c = a + b;
        a = b;
        b = c;
    }
    printf("\n");
    return 0;
}
```

Pembahasan. Hanya dua suku terakhir yang perlu disimpan (`a` dan `b`), jadi tidak perlu array. Setiap putaran mencetak `a`, lalu menggeser pasangan (a, b) menjadi $(b, a + b)$.

Soal 5.13 Rata-rata dengan Sentinel[← Soal](#)

```
#include <stdio.h>

int main() {
    long long total = 0;
    int banyak = 0, x;
    do {
        scanf("%d", &x);
        if (x != 0) {
            total += x;
            banyak++;
        }
    } while (x != 0);
    if (banyak == 0)
        printf("Tidak ada data\n");
    else
        printf("%.2f\n", (double)total / banyak);
    return 0;
}
```

Pembahasan. `do-while` cocok di sini karena input pasti dibaca minimal sekali sebelum kondisi dicek. Nilai sentinel 0 hanya menandai akhir input dan tidak boleh ikut dijumlahkan. Pembagian dengan nol dicegah saat data kosong.

Soal 5.14 Desimal ke Biner[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    long long p = 1;
    while (p * 2 <= n)
        p *= 2;
    while (p > 0) {
        printf("%lld", (n / p) % 2);
        p /= 2;
    }
    printf("\n");
    return 0;
}
```

Pembahasan. Cari dulu pangkat dua terbesar `p` yang tidak melebihi `N`, lalu cetak digit biner dari kiri ke kanan: `(n / p) % 2` adalah bit pada posisi tersebut, dan `p` dibagi 2 setiap langkah. `p` bertipe `long long` agar `p * 2` tidak meluap untuk `N` mendekati 10^9 . Untuk `N = 0`, `p` tetap 1 dan tercetak satu digit `0`.

Soal 5.15 Bilangan Sempurna[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    long long jumlah = 0;
    for (int i = 1; i < n; i++)
        if (n % i == 0)
            jumlah += i;
    if (n > 1 && jumlah == n)
        printf("SEMPURNA\n");
    else
        printf("BUKAN SEMPURNA\n");
    return 0;
}
```

Pembahasan. Jumlahkan semua `i` dari 1 sampai $N - 1$ yang membagi habis N . Untuk $N \leq 10^6$, sejuta putaran masih sangat cepat. Angka 1 bukan bilangan sempurna karena jumlah pembagi selain dirinya adalah 0.

Soal 5.16 Langkah Collatz[← Soal](#)

```
#include <stdio.h>

int main() {
    long long n;
    int langkah = 0;
    scanf("%lld", &n);
    while (n != 1) {
        if (n % 2 == 0) n /= 2;
        else n = 3 * n + 1;
        langkah++;
    }
    printf("%d\n", langkah);
    return 0;
}
```

Pembahasan. Banyaknya putaran tidak diketahui sebelumnya, jadi dipakai `while`. Untuk 6: $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$, yaitu 8 langkah. Nilai tengah bisa jauh lebih besar dari N , jadi dipakai `long long`.

Soal 5.17 Jumlah dengan Continue[← Soal](#)

```
#include <stdio.h>

int main() {
    int n, k;
    long long total = 0;
    scanf("%d %d", &n, &k);
    for (int i = 1; i <= n; i++) {
        if (i % k == 0) continue;
        total += i;
    }
    printf("%lld\n", total);
    return 0;
}
```

Pembahasan. `continue` langsung melompat ke putaran berikutnya tanpa menjalankan sisa badan perulangan. Untuk $N = 10$ dan $K = 3$, bilangan 3, 6, dan 9 dilewati: $55 - 18 = 37$.

Soal 5.18 Tebak Angka[← Soal](#)

```
#include <stdio.h>

int main() {
    int x, n, tebak, posisi = 0;
    scanf("%d %d", &x, &n);
    for (int i = 1; i <= n; i++) {
        scanf("%d", &tebak);
        if (tebak == x) {
            posisi = i;
            break;
        }
    }
    if (posisi > 0)
        printf("Benar pada tebakan ke-%d\n", posisi);
    else
        printf("Gagal\n");
    return 0;
}
```

Pembahasan. `break` menghentikan perulangan begitu tebakan benar ditemukan, sehingga yang tercatat adalah tebakan benar yang pertama. Tebakan sisanya tidak perlu dibaca. Variabel `posisi` yang tetap 0 menandakan tidak ada tebakan yang benar.

Soal 5.19 **Bilangan Armstrong**[← Soal](#)

```
#include <stdio.h>

int main() {
    int a, b, ada = 0;
    scanf("%d %d", &a, &b);
    for (int n = a; n <= b; n++) {
        int x = n / 100, y = n / 10 % 10, z = n % 10;
        if (x * x * x + y * y * y + z * z * z == n) {
            printf("%d\n", n);
            ada = 1;
        }
    }
    if (!ada) printf("Tidak ada\n");
    return 0;
}
```

Pembahasan. Periksa setiap bilangan dalam rentang: pisahkan digitnya dengan `/` dan `%`, lalu bandingkan jumlah pangkat tiganya. Variabel `ada` mencatat apakah ditemukan setidaknya satu bilangan.

Soal 5.20 **Persegi Berongga**[← Soal](#)

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            if (i == 1 || i == n || j == 1 || j == n)
                printf("*");
            else
                printf(" ");
        }
        printf("\n");
    }
    return 0;
}
```

Pembahasan. Dua perulangan bersarang menelusuri setiap sel (i, j) . Sel di baris pertama, baris terakhir, kolom pertama, atau kolom terakhir adalah tepi dan diisi `*`, sisanya spasi. Karena kolom terakhir selalu `*`, tidak ada spasi di akhir baris.