

8.1 Class Person

Description

Buatlah program yang mengimplementasikan sebuah class bernama Person, yang memiliki atribut nama (string), usia (int), tinggi (int), dan berat (double). Default constructor class ini akan secara otomatis mengisi atribut nama dengan string kosong, dan atribut lainnya diisi dengan nilai 0 (nol). Class Person memiliki prosedur bernama setPerson untuk mengisi nilai setiap atribut. Tambahkan prosedur lainnya agar dapat memproses apa yang diminta oleh program utama (main) yang diuraikan di bawah ini.

Program akan membaca n baris data yang berisi nama, usia, tinggi badan, dan berat badan. Gunakan class Person untuk menyimpan n data tersebut. Selanjutnya program akan menuliskan seluruh nama dan usia, dan dilanjutkan dengan menuliskan rata-rata tinggi badan, serta banyaknya orang (person) yang tinggi badannya di atas nilai rata-rata. Program harus benar-benar mengimplementasikan class bagi obyek Person. Oleh karena itu, tidak diperkenankan mengolah data yang berasal dari variabel biasa (bukan atribut dari class). Semua atribut class dikelompokkan sebagai private.

Input Format

Baris pertama adalah sebuah bilangan bulat n, $1 \leq n \leq 100$, yang menunjukkan banyaknya baris data yang akan dibaca. Dan n baris berikutnya berisi data nama, usia, tinggi badan, dan berat badan yang masing-masing dipisahkan oleh satu spasi. Nama orang berupa string dengan hanya terdiri atas satu kata.

Output Format

Daftar nama orang dan usia yang dipisahkan oleh satu spasi, diikuti oleh nilai rata-rata tinggi badan(dengan dua digit di belakang tanda desimal), dan banyaknya orang dengan tinggi badan di atas rata-rata. Output program diakhiri dengan newline.

Sample Input

```
4
Dudung 20 165 60.5
Abdullah 21 160 70.2
Sempurna 19 170 80.1
Oskadon 17 155 90.0
```

Sample Output

```
Dudung 20
Abdullah 21
Sempurna 19
Oskadon 17
162.50
2
```

```
1 import java.util.Scanner;
2
3 public class ClassPerson {
4     private String nama;
5     private int usia, tinggi;
6     private double berat;
7
8     public ClassPerson() {
9         nama = "";
10        usia = 0;
11        tinggi = 0;
12        berat = 0;
13    }
14
15    public void setPerson(String nama, int usia, int tinggi, double berat) {
16        this.nama = nama;
17        this.usia = usia;
18        this.tinggi = tinggi;
19        this.berat = berat;
20    }
21
22    public String getNama() {
23        return nama;
24    }
25
26    public int getUsia() {
27        return usia;
28    }
29
30    public int getTinggi() {
31        return tinggi;
32    }
33
34    public double getBerat() {
35        return berat;
36    }
37
38    public double getIMT() {
39        return berat / (tinggi * tinggi) * 10000;
40    }
41
42    public String getStatusGizi() {
43        double imt = getIMT();
44        if (imt < 17) {
45            return "sangat kurus";
46        } else if (imt < 18.5) {
47            return "kurus";
48        } else if (imt < 25) {
49            return "normal";
50        } else if (imt < 28) {
51            return "gemuk";
52        } else {
53            return "sangat gemuk";
54        }
55    }
56
57    public static void main(String[] args) {
58        Scanner sc = new Scanner(System.in);
59        double totalTinggi = 0;
60        int n;
61        n = sc.nextInt();
62        ClassPerson[] p = new ClassPerson[n];
63        for (int i = 0; i < n; i++) {
64            p[i] = new ClassPerson();
65            String nama = sc.next();
66            int usia = sc.nextInt();
67            int tinggi = sc.nextInt();
68            double berat = sc.nextDouble();
69            p[i].setPerson(nama, usia, tinggi, berat);
70            totalTinggi += p[i].getTinggi();
71        }
72        int tinggiDiatasRata = 0;
73        int statusGizi[] = {0, 0, 0, 0, 0};
74        for (int i = 0; i < n; i++) {
75            System.out.println(p[i].getNama() + " " + p[i].getUsia());
76            if (p[i].getTinggi() > totalTinggi / n) {
77                tinggiDiatasRata++;
78            }
79            if (p[i].getStatusGizi().equals("sangat kurus")) {
80                statusGizi[0]++;
81            } else if (p[i].getStatusGizi().equals("kurus")) {
82                statusGizi[1]++;
83            } else if (p[i].getStatusGizi().equals("normal")) {
84                statusGizi[2]++;
85            } else if (p[i].getStatusGizi().equals("gemuk")) {
86                statusGizi[3]++;
87            } else {
88                statusGizi[4]++;
89            }
90        }
91        System.out.printf("%.2f\n", totalTinggi / n);
92        System.out.println(tinggiDiatasRata);
93    }
94 }
```

8.2 Bilangan Pecahan

Description

Buatlah program untuk mengelola bilangan pecahan $a/b/c$ dimana a , b , dan c semuanya berupa bilangan bulat positif. Bilangan pecahan tersebut dapat diolah dengan menggunakan beberapa operasi seperti format berikut:

- set a b c inisialisasi bilangan pecahan $a/b/c$
- p menampilkan bilangan pecahan sesederhana mungkin
- add a b c menambah bilangan pecahan yang ada dengan $a/b/c$
- mul a b c mengalikan bilangan pecahan yang ada dengan $a/b/c$
- inc postfix increment
- dec postfix decrement
- end akhir dari pengolahan

Bilangan pecahan dituliskan sesederhana mungkin, dan program harus mengimplementasikan konsep OOP serta penggunaan operator overloading. Program yang tidak mengimplementasikan konsep OOP atau tidak menggunakan operator overloading, akan diberi nilai 0 (nol).

Input Format

Beberapa baris perintah sesuai deskripsi soal.

Output Format

Beberapa baris bilangan pecahan hasil operasi yang diberikan, yang dituliskan sesederhana mungkin. Setiap baris output diakhiri dengan newline.

Sample Input

```
set 4 2 8
p
add 0 6 8
inc
p
end
```

Sample Output

```
4 1/4
6
```

```
1 import java.util.*;
2 public class Pecahan {
3     private int a;
4     private int b;
5     private int c;
6     private int fpb (int m, int n) {
7         while (n!=0) {
8             int t=m%n;
9             m=n;
10            n=t;
11        }
12        return m;
13    }
14    private void simple() {
15        if (c<=0) c=1;
16        a=a+b/c;
17        b=b%c;
18        int t=fpb(b,c);
19        b=b/t;
20        c=c/t;
21    }
22    public Pecahan () {
23        a=b=0;
24        c=1;
25    }
26    public void set (int pa, int pb, int pc) {
27        a=pa;
28        b=pb;
29        c=pc;
30        simple();
31    }
32    public void add (int pa,int pb, int pc) {
33        a=a+pa;
34        b=b*pc+c*pb;
35        c=c*pc;
36        simple();
37    }
38    public void mul (int pa, int pb, int pc) {
39        if (pc<=0) pc=1;
40        a=0;
41        b=(a*c+b)*(pa*pc+pb);
42        c=1;
43        simple();
44    }
45    public void inc() {
46        add(1, 0, 1);
47    }
48    public void dec() {
49        add(-1, 0, 1);
50    }
51    public void show() {
52        if (b==0) System.out.println(a);
53        else if (a==0) System.out.printf("%d/%d\n",b,c);
54        else System.out.printf("%d %d/%d\n",a,b,c);
55    }
56}
57
58 public static void main (String[] args) {
59     Pecahan obj = new Pecahan();
60     String instruksi;
61     Scanner inp = new Scanner(System.in);
62     instruksi = inp.next();
63     int pa,pb,pc;
64     while (!instruksi.equals("end")) {
65         if (instruksi.equals("set")) {
66             pa=inp.nextInt();
67             pb=inp.nextInt();
68             pc=inp.nextInt();
69             obj.set(pa,pb,pc);
70         }
71         else if (instruksi.equals("add")) {
72             pa = inp.nextInt();
73             pb = inp.nextInt();
74             pc = inp.nextInt();
75             obj.add(pa, pb, pc);
76         } else if (instruksi.equals("mul")) {
77             pa = inp.nextInt();
78             pb = inp.nextInt();
79             pc = inp.nextInt();
80             obj.mul(pa, pb, pc);
81         } else if (instruksi.equals("inc")) {
82             obj.inc();
83         } else if (instruksi.equals("dec")) {
84             obj.dec();
85         } else if (instruksi.equals("p")) {
86             obj.show();
87         }
88         instruksi = inp.next();
89     }
90 }
91 }
92 }
93 }
```

9.1 Java ArrayList

Description

STL vector dalam C++ sangat bermanfaat untuk mengimplementasikan array dinamis. Kita tidak dipusingkan dengan ukuran array, tipe data elemen array, dan sebagainya. Dalam Java, STL Vector ini diimplementasikan dalam Java ArrayList

Gunakan Java ArrayList ini untuk menyelesaikan problem: membaca beberapa bilangan bulat, dan menghapus beberapa bilangan pada posisi tertentu. Program harus sepenuhnya menerapkan prinsip-prinsip dasar OOP.

Input Format

Bagian pertama berisi beberapa bilangan bulat yang diakhiri dengan nilai -9 (nilai ini tidak ikut dalam pengolahan, hanya sebagai akhir data masukan). Bagian kedua adalah beberapa bilangan terurut dari kecil ke besar yang menunjukkan elemen keberapa dari bilangan masukan yang dihapus. Input bagian ini juga diakhiri dengan -9 (tidak ikut diolah)

Output Format

Baris pertama adalah dua bilangan yang menunjukkan banyaknya bilangan awal dan banyaknya bilangan setelah dihapus beberapa elemennya. Baris kedua adalah dua nilai rata-rata yang dituliskan dalam dua digit di belakang tanda desimal dari bilangan-bilangan awal dan bilangan-bilangan setelah dihapus. Jika data dalam array kosong, maka nilai rata-rata dituliskan -9.99. Output program diakhiri dengan newline.

Sample Input

```
10 20 30 40 50 -9 1 3 4 -9
```

Sample Output

```
5 2
30.00 35.00
```



```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Bilangan {
5     private int jml = 0;
6     private int awal = 0;
7     private int akhir = 0;
8     private double rataawal = 0;
9     private double rataakhir = 0;
10    private ArrayList<Integer> array = new ArrayList<Integer>();
11    private Scanner input = new Scanner(System.in);
12
13    public void baca() {
14        int angka = input.nextInt();
15        while (angka != -9) {
16            array.add(angka);
17            jml += angka;
18            awal++;
19            angka = input.nextInt();
20        }
21        if (awal > 0) {
22            rataawal = (double) jml / awal;
23        }
24    }
25    public void hapus() {
26        akhir = awal;
27        int index = input.nextInt();
28        while (index != -9) {
29            jml -= array.get(index - 1);
30            akhir--;
31            index = input.nextInt();
32        }
33        if (akhir > 0) {
34            rataakhir = (double) jml / akhir;
35        }
36    }
37    public void cetak() {
38        System.out.println(awal + " " + akhir);
39        if (awal > 0) {
40            System.out.printf("%.2f ", rataawal);
41        } else {
42            System.out.print("-9.99 ");
43        }
44        if (akhir > 0) {
45            System.out.printf("%.2f\n", rataakhir);
46        } else {
47            System.out.print("-9.99\n");
48        }
49    }
50    public static void main(String[] args) {
51        Bilangan bilangan = new Bilangan();
52        bilangan.baca();
53        bilangan.hapus();
54        bilangan.cetak();
55    }
56 }
57
```

9.2 Hotspot

Description

Area titik kebakaran (Hot Spot) di Pulau Kalimantan digambarkan dalam bentuk peta grid (petak-petak) dimana setiap petak diberi koordinat kartesian XY dengan titik pusat (0,0) terletak di tengah-tengah Pulau Kalimantan seperti ilustrasi gambar berikut.

Data setiap koordinat hot spot dicatat di kertas secara manual kemudian dimasukkan ke dalam komputer untuk diolah. Karena petugas yang survei lebih dari satu orang, maka koordinat titik sering dicatat berulang. Koordinat titik yang sama akan dianggap sebagai satu titik. Buatlah program untuk membaca hasil pencatatan titik koordinat hot spot tersebut dan menuliskan koordinat titik yang berada di kuadran I dan II serta tidak berada di sumbu horizontal X dengan urutan mulai dari ujung kanan atas hingga kiri bawah. Titik A(x₁,y₁) disebut terletak di kanan atas dari titik B(x₂,y₂) jika:

- $x_1 > x_2$, atau
- $x_1 == x_2 \ \&\& \ y_1 > y_2$

Input Format

[N, 0<N<10000] banyaknya titik hotspot
[X Y] koordinat titik sebanyak N

Output Format

Baris pertama adalah banyaknya titik kebakaran. Baris kedua adalah koordinat titik dengan format seperti pada contoh. Setiap baris output diakhiri dengan newline.

Sample Input

10
1 2 1 2 -3 4 5 6 6 -5 10 11 20 -1 1 20 10 11 5 6

Sample Output

7
(10,11)(5,6)(1,20)(1,2)(-3,4)

```
1 import java.util.*;
2
3 public class Hotspot {
4     private TreeSet<Partner> titik;
5     private int nTitik;
6
7     public Hotspot () {
8         titik = new TreeSet<>(new Compare());
9         nTitik=0;
10    }
11
12    public void add(int x, int y) {
13        Partner p = new Partner(x, y);
14        titik.add(p);
15    }
16
17    public void print() {
18        int count = 0;
19        Iterator<Partner> ptr = titik.iterator();
20        while (ptr.hasNext()) {
21            ptr.next();
22            count++;
23        }
24        System.out.println(count);
25        ptr = titik.iterator();
26        while (ptr.hasNext()) {
27            Partner p = ptr.next();
28            if (p.second > 0) {
29                System.out.print("(" + p.first + "," + p.second + ")");
30            }
31        }
32        System.out.println();
33    }
34
35    public static void main(String[] args) {
36        Hotspot hotspot = new Hotspot();
37        Scanner input = new Scanner(System.in);
38        int n = input.nextInt();
39        for (int i = 0; i < n; i++) {
40            int x = input.nextInt();
41            int y = input.nextInt();
42            hotspot.add(x, y);
43        }
44        hotspot.print();
45    }
46 }
47
48 class Partner implements Comparable<Partner> {
49     public int first, second;
50
51     public Partner(int first, int second) {
52         this.first = first;
53         this.second = second;
54     }
55
56     public int compareTo(Partner other) {
57         if (this.first != other.first) {
58             return other.first - this.first;
59         } else {
60             return other.second - this.second;
61         }
62     }
63 }
64
65 class Compare implements Comparator<Partner> {
66     public int compare(Partner a, Partner b) {
67         if (a.first != b.first) {
68             return b.first - a.first;
69         } else {
70             return b.second - a.second;
71         }
72     }
73 }
74 }
```

10.1 Pewarisan Class Pegawai

Description

PT Covid Corona Corp (CCC) ingin mengelola data kepegawaianya. PT CCC memiliki 2 tipe pegawai, yaitu Pegawai Tetap dan Pegawai Harian. Perbedaan keduanya terletak pada sistem gaji, dimana Pegawai Tetap memiliki gaji pokok dan uang lembur, sedangkan Pegawai Harian hanya memiliki upah yang dihitung berdasarkan jumlah hari bekerja dalam sebulan. Untuk membuat sistem pengolahan data kepegawaian ini, PT CCC telah meminta konsultan membuat program aplikasi. Konsultan tersebut telah menyusun kerangka program menggunakan platform OOP Java. Sebagai base class, dibuat bernama Pegawai sebagai berikut:

```
class Pegawai
{
    protected String id;
    protected int usia;
    protected int tipe;
    protected int income;
    public Pegawai() { id=""; usia=income=0; tipe=1; }
    public void set(string pid, int u, int t ) {id=pid; usia=u; tipe=t; }
    public String getID() { return id; }
    public void show() { System.out.printf("%s %d %d\n", id, tipe, income); }
};
```

Dari bentuk class ini, dapat diketahui bahwa semua pegawai PT CCC memiliki atribut id (identitas pegawai), usia (dalam tahun), tipe (kelompok pegawai dalam bentuk kode bilangan, 1: Pegawai Tetap, 2: Pegawai Harian), dan income (gaji dalam bulan tertentu dalam satuan ribu rupiah). Dari base class ini, dibuat class turunan yaitu Tetap untuk semua pegawai tetap, dan Harian untuk semua pegawai harian. Selanjutnya dari ketiga class tersebut, dibuat program utama untuk membaca N data pegawai beserta informasi gaji. Tanpa mengubah definisi class Pegawai, buatlah program lengkap untuk membaca input data dan menghasilkan output daftar gaji pada bulan tertentu seperti pada format data dan contoh yang diberikan.

Input Format

[N : banyaknya pegawai, 1<N<100]
[N baris data pegawai : id, usia, tipe. Jika pegawai tetap, ada nilai gaji pokok.]
[T baris data gaji : id dan upah (pegawai harian) | uang lembur (pegawai tetap).]
END

Output Format

Beberapa baris data id, tipe, dan penghasilan sebulan. Kelompok pegawai tetap di bagian atas, disambung dengan kelompok pegawai harian. Urutan data sesuai dengan urutan data masuk. Setiap baris diakhiri dengan newline.

Sample Input

```
5
123456 19 1 5000
989212 21 1 6000
876523 20 2
092831 20 2
187632 19 1 5000
123456 2000
876523 1000
092831 5000
187632 4000
END
```

Sample Output

```
123456 1 7000
989212 1 6000
187632 1 9000
876523 2 1000
092831 2 5000
```

```
1 import java.util.Scanner;
2 import java.util.Vector;
3
4 class Pegawai {
5     protected String id;
6     protected int usia;
7     protected int tipe;
8     protected int income;
9     public Pegawai() { id=""; usia=income=0; tipe=1; }
10    public void set(String pid, int u, int t ) {id=pid; usia=u; tipe=t; }
11    public String getID() { return id; }
12    public void show() { System.out.printf("%s %d %d\n", id, tipe, income); }
13 }
14
15 class Tetap extends Pegawai {
16     private int lembur;
17
18     public void set(String pid, int u, int t, int i) {
19         super.set(pid, u, t);
20         income = i;
21     }
22
23     public void setLembur(int l) {
24         lembur = l;
25         income += lembur;
26     }
27 }
28
29 class Harian extends Pegawai {
30     private int upah;
31
32     public Harian() {
33         income = 0;
34     }
35     public void setUpah(int u) {
36         upah = u;
37         income = upah;
38     }
39 }
40
41 public class Pewarisan {
42     public static void main(String[] args) {
43         Scanner sc = new Scanner(System.in);
44         int n = sc.nextInt();
45         Vector<Tetap> pegawaiTetap = new Vector<Tetap>();
46         Vector<Harian> pegawaiHarian = new Vector<Harian>();
47         for (int i = 0; i < n; i++) {
48             String id;
49             int usia, tipe;
50             id = sc.next();
51             usia = sc.nextInt();
52             tipe = sc.nextInt();
53             if (tipe == 1) {
54                 int pokok;
55                 pokok = sc.nextInt();
56                 Tetap pegawai = new Tetap();
57                 pegawai.set(id, usia, tipe, pokok);
58                 pegawaiTetap.add(pegawai);
59             } else {
60                 Harian pegawai = new Harian();
61                 pegawai.set(id, usia, tipe);
62                 pegawaiHarian.add(pegawai);
63             }
64         }
65         String input;
66         input = sc.next();
67
68         while (!input.equals("END")) {
69             for (Tetap pegawai : pegawaiTetap) {
70                 if (pegawai.getID().equals(input)) {
71                     int lembur;
72                     lembur = sc.nextInt();
73                     pegawai.setLembur(lembur);
74                     break;
75                 }
76             }
77             for (Harian pegawai : pegawaiHarian) {
78                 if (pegawai.getID().equals(input)) {
79                     int upah;
80                     upah = sc.nextInt();
81                     pegawai.setUpah(upah);
82                     break;
83                 }
84             }
85             input = sc.next();
86         }
87         for (Tetap pegawai : pegawaiTetap) {
88             pegawai.show();
89         }
90         for (Harian pegawai : pegawaiHarian) {
91             pegawai.show();
92         }
93     }
94 }
```

10.2 Pagar Tanah Pak Agri

Description

Pak Agria memiliki tanah di N buah lokasi. Tanah tersebut berbentuk persegi panjang (panjang dan lebar) atau lingkaran (radius). Pak Agria memagari tanah miliknya. Akan tetapi, karena dana Pak Agria terbatas, ia hanya mampu memasang pagar di sebagian tanah miliknya. Dalam hal ini, Pak Agria hanya akan memasang pagar di tanah yang kelilingnya di atas rata-rata. Bantulah Pak Agria menghitung panjang pagar yang perlu ia sediakan. Gunakan nilai pi dari pustaka Math.PI untuk menghitung keliling lingkaran.

Input Format

Baris pertama: N

Digit pertama setiap baris adalah bentuk tanah (0 = persegi panjang, 1 = lingkaran).

Digit-digit berikutnya adalah panjang dan lebar, atau radius bergantung pada bentuk tanah.

Output Format

Sebuah bilangan dengan dua digit di belakang tanda desimal, yang menunjukkan panjang pagar yang dibutuhkan. Diakhiri dengan newline.

Sample Input

```
3
0 10.0 10.0
0 20.0 20.0
1 10.0
```

Sample Output

```
142.83
```

```
1 import java.util.Scanner;
2
3 interface Tanah {
4     public double KelilingTanah();
5 }
6
7 class Persegi implements Tanah {
8     private double panjang, lebar;
9
10    public Persegi(double p, double l) {
11        panjang = p;
12        lebar = l;
13    }
14
15    public double KelilingTanah() {
16        return 2 * (panjang + lebar);
17    }
18 }
19
20 class Lingkaran implements Tanah {
21     private double radius;
22
23     public Lingkaran(double r) {
24         radius = r;
25     }
26
27     public double KelilingTanah() {
28         return 2 * Math.PI * radius;
29     }
30 }
31
32 public class Tanah {
33     public static void main(String[] args) {
34         Scanner sc = new Scanner(System.in);
35         int n = sc.nextInt();
36         double totalKeliling = 0;
37         Tanah[] tanah = new Tanah[n];
38         for (int i = 0; i < n; i++) {
39             int bentuk = sc.nextInt();
40             if (bentuk == 0) {
41                 double p = sc.nextDouble();
42                 double l = sc.nextDouble();
43                 Persegi persegi = new Persegi(p, l);
44                 totalKeliling += persegi.KelilingTanah();
45                 tanah[i] = persegi;
46
47             } else {
48                 double r = sc.nextDouble();
49                 Lingkaran lingkaran = new Lingkaran(r);
50                 totalKeliling += lingkaran.KelilingTanah();
51                 tanah[i] = lingkaran;
52             }
53         }
54         double rata = totalKeliling / n;
55         double totkelatasrata = 0;
56         for (int i = 0; i < n; i++) {
57             if (tanah[i].KelilingTanah() > rata) {
58                 totkelatasrata += tanah[i].KelilingTanah();
59             }
60         }
61         System.out.printf("%.2f\n", totkelatasrata);
62     }
63 }
64 }
```

11.1 Interface Shape2D

Description

Suatu bidang tertutup 2 dimensi (shape 2D) dicatat sebagai kumpulan n buah edge (sisi) yang menghubungkan n buah titik (x,y). Sebagai contoh poligon yang memiliki 8 sisi akan dihubungkan oleh titik (x0,y0) sampai dengan titik (x7,y7).

Sebuah interface Shape2D dibuat untuk memfasilitasi 3 fungsi sebagai berikut:

- set, fungsi untuk memberikan nilai n (banyaknya sisi) dan semua titik dalam bentuk array dari koordinat x dan array dari koordinat y.
- print, mencetak koordinat titik dan koordinat titik akhir dalam format [x0,y0]-[x(n-1),y(n-1)].
- keliling, menghitung keliling dari poligon, yaitu total jumlah jarak Euclidean antara setiap titik.

Sekarang, buatlah program Java OOP yang membuat objek poligon dalam bentuk class Poligon yang mengimplementasi interface Shape2D. Program akan membaca data n dan koordinat titik sehingga dapat menghasilkan output seperti yang diberikan pada penjelasan di bawah ini.

Input Format

Baris pertama berisi sebuah bilangan bulat n, $3 < n$, menunjukkan banyaknya sisi poligon yang akan dibuat. Diikuti oleh n baris koordinat x dan y, masing-masing bilangan riil yang dipisahkan oleh satu spasi.

Output Format

Baris pertama adalah koordinat titik awal dan koordinat titik akhir dari poligon. Baris kedua adalah keliling poligon seperti definisi sebelumnya. Semua nilai ditulis dalam dua digit di belakang tanda desimal, dan setiap baris diakhiri oleh newline.

Sample Input

```
5
-1.5 2.67
0 2.5
3.45 0
2.512 -2.5
-0.25 -1.58
```

Sample Output

```
[-1.50,2.67]-[-0.25,-1.58]
15.78
```



```
1 import java.util.Scanner;
2
3 interface Shape2D {
4     void set(int n, double[] x, double[] y);
5     void print();
6     double keliling();
7 }
8
9 class Poligon implements Shape2D {
10    private int n;
11    private double[] x;
12    private double[] y;
13
14    public void set(int n, double[] x, double[] y) {
15        this.n = n;
16        this.x = x;
17        this.y = y;
18    }
19
20    public void print() {
21        System.out.printf("[%1$.2f,%2$.2f]-[%3$.2f,%4$.2f]%n", x[0], y[0], x[n-1], y[n-1]);
22    }
23
24    public double keliling() {
25        double totalDistance = 0.0;
26        for (int i = 0; i < n - 1; i++) {
27            double distance = Math.sqrt(Math.pow(x[i+1]-x[i], 2) + Math.pow(y[i+1]-y[i], 2));
28            totalDistance += distance;
29        }
30        double distance = Math.sqrt(Math.pow(x[0]-x[n-1], 2) + Math.pow(y[0]-y[n-1], 2));
31        totalDistance += distance;
32        return totalDistance;
33    }
34 }
35
36 class ShapeFactory {
37     public static Shape2D createShape(int n, double[] x, double[] y) {
38         Shape2D shape = new Poligon();
39         shape.set(n, x, y);
40         return shape;
41     }
42 }
43
44 public class Shape2DInterface {
45     public static void main(String[] args) {
46         Scanner sc = new Scanner(System.in);
47         int n = sc.nextInt();
48         double[] x = new double[n];
49         double[] y = new double[n];
50         for (int i = 0; i < n; i++) {
51             x[i] = sc.nextDouble();
52             y[i] = sc.nextDouble();
53         }
54         Shape2D shape = ShapeFactory.createShape(n, x, y);
55         shape.print();
56         double keliling = shape.keliling();
57         System.out.printf("%.2f%n", keliling);
58     }
59 }
60 }
```

11.1 Volume Silinder

Description

Diketahui struktur dari 3 obyek Lingkaran, Segiempat, dan Silinder sebagai berikut:
Buatlah program Java OOP untuk mengimplementasikan struktur ketiga objek tersebut
dan gunakan untuk membaca beberapa data komponen silinder dan menyajikan rata-rata
volume seluruh silinder serta banyaknya silinder yang memiliki volume di atas rata-rata.

Input Format

Baris pertama adalah N, $1 < N < 1000$, banyaknya silinder. Sebanyak N baris berikutnya
adalah komponen silinder berupa diameter lingkaran (d), lebar segiempat (w), dan tinggi
segiempat (h) yang semuanya adalah bilangan bulat dan dipisahkan oleh whitespace.

Output Format

Sebaris output yang terdiri atas nilai rata-rata volume seluruh silinder dengan 2 digit di
belakang tanda desimal, dan banyaknya silinder yang memiliki volume di atas rata-rata.
Kedua nilai dipisahkan oleh satu spasi.

Sample Input

```
5
2 3 4
1 5 2
2 8 9
1 2 5
1 2 3
```

Sample Output

```
9.74 2
```



```
1 import java.util.Scanner;
2 abstract class BangunDatar{
3     //int double string dkk untuk atribut itu sbg tipe data atau data type kalau method sbg return type
4     abstract double luas();
5 }
6
7 class Lingkaran extends BangunDatar {
8     private double diameter;
9     //constructor itu yang namanya sama dengan nama class, ketika class mau bikin object
10    public Lingkaran(double d) {
11        diameter = d;
12    }
13
14    public double luas() {
15        return Math.PI*(0.5*diameter)*(0.5*diameter);
16    }
17 }
18
19 class Segiempat extends BangunDatar {
20     private double lebar;
21     private double tinggi;
22
23     public Segiempat(double w, double h){
24         lebar = w;
25         tinggi = h;
26     }
27
28     public double luas() {
29         return lebar*tinggi;
30     }
31 }
32
33 class Silinder {
34     private Lingkaran alas;
35     private double tinggi;
36
37     public Silinder(double d, double h){
38         alas = new Lingkaran(d);
39         tinggi = h;
40     }
41
42     public double volume(){
43         return alas.luas()*tinggi;
44     }
45 }
46
47 public class Volumesilinder {
48     public static void main(String[] args) {
49         Scanner input = new Scanner(System.in);
50         int n = input.nextInt();
51         Silinder[] silinder = new Silinder[n]; // bikin array yang panjangnya n isinya silinder
52         double totalVolume = 0;
53         for (int i=0; i<n; i++) {
54             double d = input.nextDouble();
55             double w = input.nextDouble();
56             double h = input.nextDouble();
57             silinder[i] = new Silinder(d, h);
58             totalVolume += silinder[i].volume();
59         }
60         double rata = totalVolume / n;
61         int totvolatasrata = 0;
62         for (int i = 0; i < n; i++) {
63             if (silinder[i].volume() > rata) {
64                 totvolatasrata++;
65             }
66         }
67         System.out.printf("%.2f %d\n", rata, totvolatasrata);
68     }
69 }
```

12.1 Polymorphism Menggunakan Abstract Class

Description

Buat program Java dengan menggunakan polymorphism dengan abstract class untuk menyelesaikan permasalahan berikut. Diberikan masukan berupa nilai n, diikuti n buah baris, masing-masing berisi dua kemungkinan:

- baris diawali angka 1 dan kemudian terdapat dua buah bilangan desimal a dan b
- baris diawali dengan angka 2, kemudian terdapat 4 buah bilangan bulat, p q r s yang melambangkan dua buah pecahan $a = p/q$ dan $b = r/s$

Keluarkan beberapa buah baris yang berisi pecahan hasil penjumlahan $a + b$ dari indeks ke-i sampai dengan indeks ke-j. Jika masukan pada sebuah baris adalah pecahan desimal, maka keluarkan jumlah dalam bentuk desimal pula (dengan 2 digit di belakang koma). Jika masukan adalah pecahan rasional, keluarkan dalam bentuk pecahan rasional pula (dalam bentuk paling sederhana). CATATAN: Pemeriksaan program akan dilakukan secara manual.

Input Format

Baris pertama adalah sebuah bilangan bulat n yang menunjukkan banyaknya objek bilangan. Sebanyak n baris berikutnya adalah data dengan kemungkinan jenis 1 dan jenis 2 seperti pada deskripsi soal. Baris terakhir adalah indeks ke-i dan indeks ke-j yang menunjukkan awal dan akhir baris data yang ingin ditampilkan.

Output Format

Beberapa baris output sesuai deskripsi soal. Setiap baris diakhiri dengan newline.

Sample Input

```
5
1 0.3 0.25
2 1 2 1 4
2 1 3 1 1
1 1.5 3.78
2 1 2 3 4
2 5
```

Sample Output

```
3 4
4 3
5.28
5 4
```

```
1 import java.util.Scanner;
2 abstract class Bilangan {
3     public abstract void tampil();
4 }
5
6 class DuaBilangan extends Bilangan {
7     private double a, b;
8
9     public DuaBilangan(double a1, double b1) {
10        a = a1;
11        b = b1;
12    }
13
14    public void tampil() {
15        System.out.printf("%.2f\n", a+b);
16    }
17 }
18
19 class EmpatBilangan extends Bilangan {
20     private int p, q, r, s;
21
22     public EmpatBilangan(int p1, int q1, int r1, int s1) {
23         p = p1;
24         q = q1;
25         r = r1;
26         s = s1;
27     }
28
29     private int fpb (int a, int b){
30         while (b!=0) {
31             int t = a%b;
32             a = b;
33             b = t;
34         }
35         return a;
36     }
37
38     public void tampil() {
39         int a = p*s + q*r;
40         int b = q*s;
41         int t = fpb(a,b);
42         a=a/t;
43         b=b/t;
44         System.out.printf("%d %d\n", a, b);
45     }
46 }
47
48 public class PolymorphismeAbstractClass {
49     public static void main(String[] args) {
50         Scanner input = new Scanner(System.in);
51         int n = input.nextInt();
52         Bilangan obj[] = new Bilangan[n];
53         for (int i=0; i<n; i++) {
54             int jenis = input.nextInt();
55             if (jenis==1) {
56                 double a = input.nextDouble();
57                 double b = input.nextDouble();
58                 obj[i] = new DuaBilangan(a,b);
59             } else {
60                 int p = input.nextInt();
61                 int q = input.nextInt();
62                 int r = input.nextInt();
63                 int s = input.nextInt();
64                 obj[i] = new EmpatBilangan(p,q,r,s);
65             }
66         }
67         int i = input.nextInt();
68         int j = input.nextInt();
69         for (int k=i-1; k<=j-1; k++) {
70             obj[k].tampil();
71         }
72     }
73 }
74 }
```

12.2 Polymorphism Menggunakan Interface

Description

Buat program Java dengan menggunakan polymorphism dengan interface untuk menyelesaikan permasalahan berikut. Diberikan masukan berupa nilai n, diikuti n buah baris, masing-masing berisi dua kemungkinan:

- baris diawali angka 1 dan kemudian terdapat dua buah bilangan desimal a dan b
- baris diawali dengan angka 2, kemudian terdapat 4 buah bilangan bulat, p q r s yang melambangkan dua buah pecahan $a = p/q$ dan $b = r/s$

Keluarkan beberapa buah baris yang berisi pecahan hasil penjumlahan $a + b$ dari indeks ke-i sampai dengan indeks ke-j. Jika masukan pada sebuah baris adalah pecahan desimal, maka keluarkan jumlah dalam bentuk desimal pula (dengan 2 digit di belakang koma). Jika masukan adalah pecahan rasional, keluarkan dalam bentuk pecahan rasional pula (dalam bentuk paling sederhana). CATATAN: Pemeriksaan program akan dilakukan secara manual.

Input Format

Baris pertama adalah sebuah bilangan bulat n yang menunjukkan banyaknya objek bilangan. Sebanyak n baris berikutnya adalah data dengan kemungkinan jenis 1 dan jenis 2 seperti pada deskripsi soal. Baris terakhir adalah indeks ke-i dan indeks ke-j yang menunjukkan awal dan akhir baris data yang ingin ditampilkan.

Output Format

Beberapa baris output sesuai deskripsi soal. Setiap baris diakhiri dengan newline.

Sample Input

```
5
1 0.3 0.25
2 1 2 1 4
2 1 3 1 1
1 1.5 3.78
2 1 2 3 4
2 5
```

Sample Output

```
3 4
4 3
5.28
5 4
```

```
1
2 import java.util.Scanner;
3 interface Bilangan {
4     public void tampil();
5 }
6
7 class DuaBilangan implements Bilangan {
8     private double a, b;
9
10    public DuaBilangan(double a1, double b1) {
11        a = a1;
12        b = b1;
13    }
14
15    public void tampil() {
16        System.out.printf("%.2f\n", a+b);
17    }
18 }
19
20 class EmpatBilangan implements Bilangan {
21     private int p, q, r, s;
22
23    public EmpatBilangan(int p1, int q1, int r1, int s1) {
24        p = p1;
25        q = q1;
26        r = r1;
27        s = s1;
28    }
29
30    private int fpb (int a, int b){
31        while (b!=0) {
32            int t = a%b;
33            a = b;
34            b = t;
35        }
36        return a;
37    }
38
39    public void tampil() {
40        int a = p*s + q*r;
41        int b = q*s;
42        int t = fpb(a,b);
43        a=a/t;
44        b=b/t;
45        System.out.printf("%d %d\n", a, b);
46    }
47 }
48
49 public class PolymorphismeInterface {
50     public static void main(String[] args) {
51         Scanner input = new Scanner(System.in);
52         int n = input.nextInt();
53         Bilangan obj[] = new Bilangan[n];
54         for (int i=0; i<n; i++) {
55             int jenis = input.nextInt();
56             if (jenis==1) {
57                 double a = input.nextDouble();
58                 double b = input.nextDouble();
59                 obj[i] = new DuaBilangan(a,b);
60             } else {
61                 int p = input.nextInt();
62                 int q = input.nextInt();
63                 int r = input.nextInt();
64                 int s = input.nextInt();
65                 obj[i] = new EmpatBilangan(p,q,r,s);
66             }
67         }
68         int i = input.nextInt();
69         int j = input.nextInt();
70         for (int k=i-1; k<=j-1; k++) {
71             obj[k].tampil();
72         }
73     }
74 }
75 }
76 }
```

13.1 Kalkulator

Description

Suatu kalkulator akan memiliki nilai `currentValue` yang awalnya bernilai 0. Setelah itu, nilai tersebut akan dimodifikasi dengan operasi aritmatika yang diberikan terus menerus hingga kalkulator dimatikan. Pada soal ini, kalian diminta untuk membuat suatu interface yang mendefinisikan sifat suatu kalkulator yang dapat melakukan beberapa operasi matematika. Detail interface yang perlu kalian buat dapat dilihat pada deskripsi berikut:

[1] AritmatikaDasar - berisi fungsi-fungsi:

- `double tambah(double a, double b)`: menjumlahkan a dan b.
- `double kurang(double a, double b)`: mengurangi a dengan b.
- `double kali(double a, double, b)`: mengalikan a dan b;
- `double bagi(double a, double b)`: membagi a dengan b

[2] AritmatikaLanjut - berisi fungsi-fungsi:

- `double akarKuadrat(double a)`: mengembalikan nilai akar kuadrat dari a.
- `double pangkat(double a, double b)`: mengembalikan nilai a pangkat b

[3] KalkulatorSaintifik - merupakan kalkulator yang dapat melakukan operasi AritmatikaDasar dan AritmatikaLanjut. Juga memiliki fungsi `void clear()`: mengembalikan nilai `currentValue` pada kalkulator menjadi 0.

Selanjutnya buatlah objek kalkulator dengan jenis KalkulatorSaintifik yang mengolah data seperti pada penjelasan di bawah ini.

Input Format

Setiap baris masukan merupakan salah satu dari tujuh kemungkinan berikut. Masukan akan diakhiri dengan simbol ~.

- + X : menambah `currentValue` dengan X.
- - X : mengurangi `currentValue` dengan X.
- * X : mengalikan `currentValue` dengan X.
- / X : membagi `currentValue` dengan X.
- ^ X : memangkatkan `currentValue` dengan X.
- # : akar kuadrat dari `currentValue`
- C : mengembalikan nilai `currentValue` menjadi 0.
- ~ : mematikan kalkulator, program berhenti

Output Format

Nilai-nilai `currentValue` setiap kali suatu operasi selesai dilakukan. Cetak dengan dua angka di belakang koma dan akhiri dengan newline.

Sample Input

```
+ 10  
- 5  
* 2  
/ 5  
^ 2  
#  
C  
+ 5  
~
```

Sample Output

```
10.00  
5.00  
10.00  
2.00  
4.00  
2.00  
0.00  
5.00
```

```
1 import java.util.Scanner;
2
3 interface AD {
4     double tambah(double a, double b);
5     double kurang(double a, double b);
6     double kali(double a, double b);
7     double bagi(double a, double b);
8 }
9
10 interface AL {
11     double akarKuadrat(double a);
12     double pangkat(double a, double b);
13 }
14
15 interface KS extends AD, AL {
16     void clear();
17 }
18
19 public class Kalkulator implements KS {
20     public double currentValue;
21
22     public Kalkulator() {
23         currentValue = 0;
24     }
25
26     public double tambah(double a, double b) {
27         return currentValue += b;
28     }
29
30     public double kurang(double a, double b) {
31         return currentValue -= b;
32     }
33
34     public double kali(double a, double b) {
35         return currentValue *= b;
36     }
37
38     public double bagi(double a, double b) {
39         return currentValue /= b;
40     }
41
42     public double akarKuadrat(double a) {
43         return currentValue = Math.sqrt(currentValue);
44     }
45
46     public double pangkat(double a, double b) {
47         return currentValue = Math.pow(currentValue, b);
48     }
49
50     public void clear() {
51         currentValue = 0.00;
52     }
53
54     public static void main(String[] args) {
55         Scanner inp = new Scanner(System.in);
56         Kalkulator obj = new Kalkulator();
57         String opr = inp.next();
58
59         while (!opr.equals("~")) {
60             if (opr.equals("+")) {
61                 double X = inp.nextDouble();
62                 System.out.printf("%.2f\n", obj.tambah(0, X));
63             } else if (opr.equals("-")) {
64                 double X = inp.nextDouble();
65                 System.out.printf("%.2f\n", obj.kurang(obj.currentValue, X));
66             } else if (opr.equals("*")) {
67                 double X = inp.nextDouble();
68                 System.out.printf("%.2f\n", obj.kali(obj.currentValue, X));
69             } else if (opr.equals("/")) {
70                 double X = inp.nextDouble();
71                 System.out.printf("%.2f\n", obj.bagi(obj.currentValue, X));
72             } else if (opr.equals("#")) {
73                 System.out.printf("%.2f\n", obj.akarKuadrat(obj.currentValue));
74             } else if (opr.equals("^")) {
75                 double X = inp.nextDouble();
76                 System.out.printf("%.2f\n", obj.pangkat(obj.currentValue, X));
77             } else if (opr.equals("C")) {
78                 obj.clear();
79                 System.out.printf("%.2f\n", obj.currentValue);
80             }
81
82             opr = inp.next();
83         }
84     }
85 }
86 }
```

13.2 Simulasi Ujian

Description

Pada problem ini, buatlah sebuah program berorientasi objek dengan bahasa Java untuk mensimulasikan suatu ujian. Suatu ujian terdiri atas beberapa soal isian. Soal isian berisi pasangan pertanyaan dan kunci jawaban yang dapat diperiksa dengan jawaban yang diberikan. Nilai ujian diberikan dengan formula Jumlah Jawaban Benar / Total Pertanyaan.

Program yang dibuat harus memuat setidaknya kelas Soal, kelas Soallsian, dan kelas Ujian serta sebuah interface bernama Answerable yang memiliki abstract method:

```
boolean cekJawaban(String jawaban);
```

Jawaban yang tidak memuat hal-hal di atas secara lengkap akan diberi nilai 0 pada pemeriksaan manual.

Input Format

1. Baris pertama berisi jumlah pertanyaan (N) yang diberikan pada ujian (maksimal 100).
2. Dua baris berikutnya berisi pasangan pertanyaan dan kunci jawaban, dengan total N pasangan pertanyaan dan kunci jawaban.
3. Baris berikutnya adalah sebuah bilangan bulat yang berisi jumlah pertanyaan yang dijawab.
4. Dua baris berikutnya berisi nomor pertanyaan yang dijawab dan jawaban yang diberikan. Satu pertanyaan hanya bisa dijawab paling banyak satu kali, namun pertanyaan tidak harus dijawab secara berurutan.

Output Format

Sebuah bilangan bulat dengan dua digit di belakang koma yang merupakan nilai yang diperoleh. Akhiri dengan karakter newline.

Sample Input

```
3
Nama Ibukota Indonesia?
DKI Jakarta
Kampus IPB salah satunya terletak di?
Darmaga
Nama Fakultas dengan Kode G?
FMIPA
3
2
Dramaga
1
DKI Jakarta
3
FMIPA
```

Sample Output

```
66.67
```

Penjelasan

Pada contoh di atas, pertanyaan dijawab dengan urutan pertanyaan 2, 1, dan 3. Karena 2 dari 3 jawaban benar, maka nilai akhir yang diberikan adalah 66.67.

```
 1 import java.util.Scanner;
 2
 3 interface Answerable {
 4     boolean cekJawaban(String jawaban);
 5 }
 6
 7
 8 class Soal implements Answerable {
 9     private String pertanyaan;
10     private String jawaban;
11
12     public Soal (String pertanyaan, String jawaban) {
13         this.pertanyaan = pertanyaan;
14         this.jawaban = jawaban;
15     }
16
17     public boolean cekJawaban(String jawaban) {
18         return jawaban.equals(this.jawaban); // return true if jawaban adalah jawaban
19     }
20
21 }
22
23 class SoalIsian extends Soal {
24     public SoalIsian(String soalpertanyaan, String kuncijawaban) { // parameter soalpertanyaan dan kuncijawaban
25         super(soalpertanyaan, kuncijawaban);
26     }
27
28     public boolean cekJawaban(String jawaban) {
29         return super.cekJawaban(jawaban);
30     }
31
32 }
33
34 class Ujian {
35     private SoalIsian[] soalIsian;
36     private int jumlahSoal;
37     private int jumlahBenar;
38     Scanner input = new Scanner(System.in);
39
40     // method untuk menerima soal
41     public void getSoal() {
42         int n = input.nextInt();
43         soalIsian = new SoalIsian[n];
44         input.nextLine(); // untuk mengabaikan newline, karena nextInt() tidak mengabaikan newline
45         for (int i = 0; i < n; i++) {
46             String pertanyaan = input.nextLine(); // kalau next aja itu membaca sampe white space aj,
47             //klo nextLine itu membaca sampe newline pertama
48             String jawaban = input.nextLine();
49             soalIsian[i] = new SoalIsian(pertanyaan, jawaban);
50         }
51         jumlahSoal = n;
52     }
53
54     // method untuk menerima jawaban
55     public void getJawaban() {
56         int n = input.nextInt();
57         input.nextLine(); // untuk mengabaikan newline, karena nextInt() tidak mengabaikan newline
58         for (int i = 0; i < n; i++) {
59             int nomor = input.nextInt();
60             input.nextLine();
61             String jawaban = input.nextLine();
62             if (soalIsian[nomor-1].cekJawaban(jawaban)) {
63                 jumlahBenar++;
64             }
65         }
66     }
67
68     // method untuk menampilkan nilai
69     public void getNilai() {
70         double nilai = (double) jumlahBenar / jumlahSoal * 100;
71         System.out.printf("%.2f\n", nilai);
72     }
73
74 }
75 public class SimulasiUjian {
76     public static void main(String[] args) {
77         Ujian ujian = new Ujian();
78         ujian.getSoal();
79         ujian.getJawaban();
80         ujian.getNilai();
81     }
82 }
83 }
```