

{ ----- 1 ----- }

var x: array[1..100] of real;

avg, sum:real;

i: integer;

begin

for i:=1 to 100 do

begin

read(x[i]);

sum := sum+x[i];

end;

avg:=sum/100;

for i:=1 to 100 do

if x[i]>avg then

writeln(x[i])

end.

{ ----- 2 ----- }

type matrix = array[1..5, 1..6] of real;

function neg(x: matrix):integer;

var n, i, j:integer;

n:=0;

for i:=0 to 5 do

for j:=0 to 6 do

if x[i, j]<0 then

n:=n+1;

neg:=n

end;

{ ----- 3 ----- }

var name: array[1..30] of string;

avg: array[1..30] of real;

```

tn: string;
ta: real;
i, j: integer;
begin
  for i:=1 to 30 do
    begin
      readln(name[i]);
      readln(avg[i])
    end;

    for i:=29 downto 1 do
      for j:=1 to i do
        if avg[j]<avg[j+1] then
          begin
            ta:=avg[j];
            avg[j]:=avg[j+1];
            avg[j+1]:=ta;
            tn:=name[j];
            name[j]:=name[j+1];
            name[j+1]:=tn
          end;
        end;
      end;
    end;

    for i:=1 to 30 do
      writeln(name[i])
    end.

{ ----- 4 ----- }
type material=record
  name:string;
  m, v:real
end;

```

```
function density(x: material):real;
begin
    density:= x.m/x.v
end;
```

```
{ ----- 5 ----- }
type arrmat=array[1..50] of material;
```

```
function summ(var x: arrmat):real;
var s:real;
    i: integer;
begin
    s:=0;
    for i:=1 to 50 do
        s:=s+x[i].m;
    summ:=s
end;
```

```
{ ----- 6 ----- }
var a: arrmat;
    i: integer;
    minden:real;
    minname:string;
begin
    minden:=1e38;
    for i:=1 to 50 do
        with a[i] do
            begin
                readln(name);
                readln(m, v);
```

```

        if density(a[i])<minden then
        begin
            minden:= density(a[i]);
            minname:=name
        end
    end;

```

```

        writeln(minname)

```

```

end.

```

```

{ ----- 7 ----- }

```

```

type matrix=array[1..5, 1..4] of real;

```

```

procedure tarsim(x:matrix);

```

```

var i, j: integer;

```

```

begin

```

```

    for i:=1 to 5 do

```

```

        begin

```

```

            writeln;

```

```

            for j:=1 to 4 do

```

```

                writeln(x[i,j]:6:2)

```

```

            end

```

```

        end;

```

```

{ ----- 8 ----- }

```

```

var x:array[1..100] of integer;

```

```

    i, j: integer;

```

```

begin

```

```

    for i:=1 to 100 do

```

```

        read(x[i]);

```

```

for i:=1 to 100 do
begin
    for j:=1 to i-1 do
        if x[j]=x[i] then
            break;

    if x[j]=x[i] then
begin
    writeln(x[i]);
    break
end
end
end
end.

{ ----- 9 ----- }
type arr=array[1..100] of integer;

```

```

function ascending(x: arr): boolean;
var i: integer;
    t:boolean;
begin
    t:=true;
    for i:=2 to 100 do
        if x[i]<x[i-1] then
begin
            t:=false;
            break
end;
end;

```

```

    ascending:=t
end;

{ ----- 10 ----- }
type article=record
    name: string;
    pbuy, psale:integer;
    weight: real
end;

{ ----- 11 ----- }
function interest(x:article): real;
begin
    interest:= (x.psale-x.pbuy)/x.pbuy*100
end;

{ ----- 12 ----- }
var a: array[1..50] of article;
    i, ig, nsud: integer;
    sumw: real;
begin
    ig:=1;
    for i:=1 to 50 do
        with a[i] do
            begin
                readln(name);
                readln(pbuy, psale, weight);
                sumw:=sumw+weight;
                if psale> a[ig].psale then
                    ig:=i;
            end
        end
    end
end

```

```
    if interest(a[i])>25 then
```

```
        nsud:=nsud+1
```

```
    end;
```

```
    writeln(sumw); { a }
```

```
    writeln(a[ig].name); { b }
```

```
    writeln(nsud) { c }
```

```
end.
```

```
{ ----- 13 ----- }
```

```
var i:integer;
```

```
    s:real;
```

```
begin
```

```
    for i:=1 to 99 do
```

```
        s:= s+ (i + i+1)/(i*(i+1));
```

```
    writeln(s)
```

```
end.
```

```
{ ----- 14 ----- }
```

```
var x, y, i, n: integer;
```

```
begin
```

```
    read(x, y);
```

```
    for i:=1 to x do
```

```
        if (x mod i=0) and (y mod i = 0) then
```

```
            n:=n+1;
```

```
    writeln(n)
```

```
end.
```

```
{ ----- 15 ----- }
```

```

var x, s, n: integer;
begin
  while true do
  begin
    read(x);
    if x<0 then
      break;

    if x mod 2 = 1 then
    begin
      s:=s+x;
      n:=n+1
    end
  end;

  writeln(s/n)
end.

```

```

{ ----- 16 ----- }
var x: array[1..4, 1..5] of real;
    nsefr, i, j: integer;
begin
  for i:=1 to 4 do
    for j:=1 to 5 do
      begin
        read(x[i, j]);
        if x[i, j]=0 then
          nsefr:=nsefr+1
        end;
      end;
    end;
  end;

  writeln(nsefr); { a }

```

```

for i:=1 to 4 do { b }
begin
    writeln;
    for j:=1 to 5 do
        write(x[i, j]:6:2)
    end
end.

{ ----- 17 ----- }
type fbclub=record
    name:string;
    won, loss, equ, gavg: integer
end;

{ ----- 18 ----- }
procedure getfc(var x:fbclub);
begin
    with x do
    begin
        readln(name);
        readln(won, loss, gavg)
    end
end;

{ ----- 19 ----- }
function fcscor(x:fbclub):integer;
begin
    fcscor:=x.won*3+x.equ
end;

```

{ ----- 20 ----- }

function fccompare(a, b: fbclub):boolean;

begin

if fcscor(a) = fcscor(b) then

fccompare := a.gavg>b.gavg

else

fccompare:= fcscor(a) > fcscor(b)

end;

{ ----- 21 ----- }

var a: array[1..18] of fbclub;

i, ichamp: integer;

begin

ichamp:=1;

for i:=1 to 18 do

begin

getfc(a[i]);

if fccompare(a[i], a[ichamp]) then

ichamp:=i

end;

writeln(a[ichamp].name)

end.