

```

{test program for subset pascal compiler }
{"saishou jijyouhou no keisann}

program pas104(output);

var
  n:integer;
  max: integer;   { const for the number of data }
  sx:integer;
  sy:integer;
  sxy:integer;
  sx2:integer;
  w:integer;
  a:integer;
  b:integer;
  dx:array [1..8] of integer;
  dy:array [1..8] of integer;

begin
  max := 8;
  dx[1]:=96;   dy[1]:=86;
  dx[2]:=89;   dy[2]:=56;
  dx[3]:=78;   dy[3]:=81;
  dx[4]:=68;   dy[4]:=86;
  dx[5]:=58;   dy[5]:=78;
  dx[6]:=49;   dy[6]:=56;
  dx[7]:=39;   dy[7]:=23;
  dx[8]:=32;   dy[8]:=24;

  n:=1;sx:=0;sy:=0;sxy:=0;sx2:=0;
  writeln('No. x    y    xy    x^2');
  while n<=max do begin
    writeln(n, ' ', dx[n], ' ', dy[n], ' ',
            dx[n]*dy[n], ' ', dx[n]*dx[n]);
    sx:=sx+dx[n];   sy:=sy+dy[n];
    sxy:=sxy+dx[n]*dy[n];writeln('sxy = ',sxy);
    sx2:=sx2+dx[n]*dx[n];writeln('sx2 = ',sx2);
    n:=n+1
  end;
  writeln;
  writeln( 'Sigma(x)= ',sx);
  writeln( 'Sigma(y)= ',sy);
  writeln( 'Sigma(xy)= ',sxy);
  writeln( 'Sigma(x^2)= ',sx2);
  w:=max*sx2-sx*sx;
  if w*sx2=0 then begin
    writeln( 'Fitting Unsuccessful.')
  end
  else begin
    b:=(sx2*sy - sxy*sx) div w;
    a:=(sxy-b*sx) div sx2;
    writeln;
    writeln( 'a=',a,'    b=',b)
  end
end.

```

```

PROG  START  BEGIN

BEGIN
; begin assign
LEA  GR2,VAR1
PUSH 0,GR2
LEA  GR1,8
POP   GR2
ST   GR1,0,GR2 ; end assign
; begin assign
LEA  GR1,1
LEA  GR2,-1,GR1
LEA  GR2,VAR9,GR2
}

LD   GR1,0,GR2
POP  GR2
CALL MULT
LEA  GR1,0,GR2
SUB  GR1,0,GR4
MUL  GR1,MINUS
POP  GR0
POP  GR2
ST   GR1,0,GR2 ; end assign
; begin if
LEA  GR2,VAR6 } W x 2 (GR1)
LD   GR1,0,GR2 }
PUSH 0,GR1      stack top 1 左のレジスタに 0 を
LEA  GR2,VAR5 } S x 2 x 2
LD   GR1,0,GR2 }
POP  GR2
CALL MULT      GR1 x GR2 -> GR2
LEA  GR1,0,GR2 GR2 -> GR1
PUSH 0,GR1     GR1 を 0
LEA  GR1,0     0 代入
CPA  GR1,0,GR4 0 と 0 を比較
POP  GR0      0 と 0 を比較した結果を比較
JNZ  F2      0 と 0 を比較した結果が 0 ではない
T2   LEA  GR1,1  True = 1 を代入
      JMP  B2
F2   LEA  GR1,0  False = 0 を代入
B2
CPA  GR1,MASK=1 True か 0 を比較
JNZ  ELO      False を代入
; begin write
LEA  GR1,21
LEA  GR2,ST11 ; 'Fitting Unsuccessful.'
CALL WRTSTR
CALL WRITLN ; end write
JMP  ENO

ELO
; begin assign
LEA  GR2,VAR8
PUSH 0,GR2
LEA  GR2,VAR5
LD   GR1,0,GR2
PUSH 0,GR1
LEA  GR2,VAR3
LD   GR1,0,GR2
POP  GR2
CALL MULT
LEA  GR1,0,GR2
PUSH 0,GR1

```

```

LEA    GR2,ST12    ; 'a='
CALL   WRTSTR
LEA    GR2,VAR7
LD     GR1,0,GR2
LEA    GR2,0,GR1
CALL   WRTINT
LEA    GR1,5
LEA    GR2,ST13    ; ' b='
CALL   WRTSTR
LEA    GR2,VAR8
LD     GR1,0,GR2
LEA    GR2,0,GR1
CALL   WRTINT
CALL   WRTLN      ; end write
ENO    ; end if
EXIT

```

```
VAR0   DS      1
```

```

VAR1   DS      1
VAR2   DS      1
VAR3   DS      1
VAR4   DS      1
VAR5   DS      1
VAR6   DS      1
VAR7   DS      1
VAR8   DS      1
VAR9   DS      8
VAR17  DS      8
ST0    DC      'No. x    y    xy    x^2'
ST1    DC      ' '
ST2    DC      ' '
ST3    DC      ' '
ST4    DC      ' '
ST5    DC      'sxy = '
ST6    DC      'sx2 = '
ST7    DC      'Sigma(x)= '
ST8    DC      'Sigma(y)= '
ST9    DC      'Sigma(xy)= '
ST10   DC      'Sigma(x^2)= '
ST11   DC      'Fitting Unsuccessful.'
ST12   DC      'a='
ST13   DC      ' b='
MASK   DC      1
MINUS  DC      -1
END

```

3

```
program z20(input, output);
var x, y, Result: integer;

procedure gcd(x, y: integer);
var nextY, tmp: integer;
begin
  if x < y then
  begin
    tmp := x;
    x := y;
    y := tmp;
  end;
  if y = 0 then
  begin Result := x end
  else begin
    nextY := x mod y;
    gcd(y, nextY)
  end
end;

begin
  writeln('Input x');
  readln(x);
  if x >= 0 then begin
    writeln('Input y');
    readln(y);
    if y >= 0 then
    begin
      gcd(x, y);
      writeln('gcd(' , x, ', ', y, ') = ', Result)
    end
    else begin
      writeln('Input error')
    end
  end
  else begin
    writeln('Input error')
  end
end
end.
```

```

; begin assign
LEA GR2,VAR2
PUSH 0,GR2
LEA GR2,0
ADD GR2,0,GR3
LD GR1,0,GR2
POP GR2
ST GR1,0,GR2 ; end assign
JMP EN1

```

EL1

```

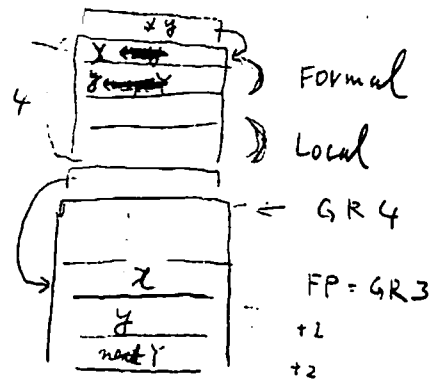
; begin assign
LEA GR2,2
ADD GR2,0,GR3
PUSH 0,GR2
LEA GR2,0
ADD GR2,0,GR3
LD GR1,0,GR2
PUSH 0,GR1
LEA GR2,1
ADD GR2,0,GR3
LD GR1,0,GR2
POP GR2
CALL MOD
LEA GR1,0,GR2
POP GR2
ST GR1,0,GR2 ; end assign

```

```

; begin gcd
PUSH 0,GR3
LEA GR4,-4,GR4
LEA GR2,1
ADD GR2,0,GR3
LD GR1,0,GR2
ST GR1,0,GR4
LEA GR2,2
ADD GR2,0,GR3
LD GR1,0,GR2
ST GR1,1,GR4
PUSH 0,GR4
LEA GR3,0,GR4
CALL PRO1 ; call gcd
LEA GR4,5,GR3
POP GR3 ; end gcd
EN1 ; end if
RET

```



BEGIN

```

; begin write
LEA GR1,7
LEA GR2,ST0 ; 'Input x'
CALL WRTSTR
CALL WRTLN ; end write
; begin read
LEA GR2,VAR0
CALL RDINT
CALL RDLN ; end read
; begin if
LEA GR2,VAR0
LD GR1,0,GR2
PUSH 0,GR1
LEA GR1,0
CPA GR1,0,GR4
POP GR0
JZE T3
JPZ F3
T3 LEA GR1,1
JMP B3

```

T3

5