```
//Scanf and Print function for multiple digit
    ;An assembly program to read 16 bit input
;and print that number again from and to consol

ORG 100H

MAIN PROC
   START:

   CALL SCAN
   CALL PRINT

   ;Print a new line
   MOV AH, 2
   MOV DL, 10
   INT 21H
   MOV DL, 13
   INT 21H
   JMP START   ;Take input again
   RET
ENDP

;A procedure that reads a 16 bit signed input
;and store that in AX
SCAN PROC
   ;Backup register values in stack
   PUSH BX
   PUSH CX
   PUSH DX

;Clear register values
XOR BX, BX
XOR CX, CX

;Read first character
MOV AH, 1
INT 21H

;Check if it is a sign or digit
CMP AL, '-'
JE NEGATIVE
CMP AL, '+'
JE POSITIVE
JMP INPUT

NEGATIVE:
;Store that it is negative number in CX
MOV CX, 1

POSITIVE:
;Take a digit input if first input is sign
INT 21H

INPUT:
;Convert the digit ASCII to number
AND AX, 000FH
;As multiplication erases value in AX
```

```
    ;backup the digit to stack
    PUSH AX
    ;Multiply previous value by 10 and add new value
    MOV AX, 10
    MUL BX
    ;Pop new digit from stack
    POP BX
    ADD BX, AX
    ;Read digit repeatedly until space or carriage return read
    MOV AH, 1
    INT 21H
    CMP AL, ' '
    JE ENDINPUT
    CMP AL, 13
    JE CARRIAGERETURN
    JMP INPUT

    CARRIAGERETURN:
    ;If last input is carriage return, print a new line
    MOV AH, 2
    MOV DL, 10
    INT 21H

    ;Store the positive input to AX
    ENDINPUT:
    MOV AX, BX

    ;Check if the value is negative
    CMP CX, 0
    JE ENDSCAN
    NEG AX

    ENDSCAN:
    ;Restore register values from stack
    POP DX
    POP CX
    POP BX
    RET
ENDP

; A procedure that prints number stored in AX in decimal format
PRINT PROC
    ;Backup register values in stack
    PUSH AX
    PUSH BX
    PUSH CX
    PUSH DX

    ;Check if Ax is positive or negative
    CMP AX, 0
    JGE INIT

    PUSH AX
    MOV DL, '-'
    MOV AH, 2
```

```
INT 21H
POP AX
NEG AX

INIT:
XOR CX, CX  ;Clear CX. Holds number of digits
MOV BX, 10  ;Holds divisor

DIGITIFY:
CWD        ;Clear DX
DIV BX
PUSH DX     ;Push last digit to stack
INC CX

;Check if the quotient is zero
CMP AX, 0
JNZ DIGITIFY

;Pop and print
MOV AH, 2
PRINTLOOP:
POP DX
OR DL, 30H  ;Convert to ASCII
INT 21H
LOOP PRINTLOOP

;Restore register values from stack
POP DX
POP CX
POP BX
POP AX
RET
ENDP
```

Uva prblm solved by c and assembly######

10035 Uva problem solved by using c:

```
#include<stdio.h>
int main()
{
   int a,b,r1,r2;
   while(1)
   {
int sum=0,c=0;
      scanf("%d%d",&a,&b);
      if(a==0&&b==0)
         break;
      while(a>0&&b>0)
      {


         r1=a%10;


         r2=b%10;


         sum = r1+r2;
         a=a/2;
         b=b/2;
         if(sum+c>=10)
         {
            c++;
         }
      }
      if(c>0)
      {
         printf("%d carry operation\n",c);
      }
      else
         printf("No carry operation \n");
   }
   return 0;
}
```

10035 Uva problem solved by using assembly:

```
;UVA PRBLM 10035


.MODEL SMALL
.STACK 100H
.DATA
SUM DW 0
A DW 0
B DW 0
R1 DW 0
R2 DW 0
C dw ?
D DW 0
E DW 0
P DW 0
.CODE
MAIN PROC
```

```
STRT:
MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
MOV DS,AX

 INCLUDE 'EMU8086.INC'


XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

MOV D,10
MOV E,2

CALL SCAN_NUM
;ADD P,CX
MOV A,CX

PRINTN

CALL SCAN_NUM
; ADD Q,CX
MOV B,CX
PRINTN

CMP A,0
JE EXIT

CMP B,0
JE EXIT

WHILE:
CMP A,0
JG CHECK

CHECK:
  CMP B,0
  JG CAL
  JMP IF

CAL:
   XOR AX,AX
   ADD AX,A
   DIV D
   MOV R1,DX
   XOR DX,DX  ;R1
   XOR AX,AX

   ADD AX,B
   DIV D
   MOV R2,DX ;R2
   XOR DX,DX
   XOR AX,AX

   MOV AX,R1   ;R2+R1
   ADD AX,R2
   MOV SUM,AX
```

```
                                        JMP WHILE
XOR AX,AX

                                      IF:
MOV AX,A                               CMP C,0
                                       JG PRINT1
DIV E
MOV A,AX                              PRINT1:
 XOR AX,AX   ;A/2                       MOV AX,C
XOR DX,DX                              CALL PRINT_NUM

                                       PRINTN "CARRRY OPERATION"
MOV AX,B
DIV E                                  JMP EXIT
MOV B,AX
XOR AX,AX                             PRINT2:
XOR DX,DX  ;B/2                        PRINTN " NO CARRY OPERATION"

MOV AX,SUM
ADD AX,C                              EXIT:

CMP AX,10                              MOV AH,4CH
JGE L2                                 INT 21H
JMP WHILE                             MAIN ENDP
                                      DEFINE_SCAN_NUM
L2:                                    DEFINE_PRINT_NUM
                                       DEFINE_PRINT_NUM_UNS
 INC C                                  END MAIN
```

```
//UVA PRBLM 10079

#include<stdio.h>
int main()
{
   int p;
   long long int n;
   while(scanf("%lld",&n))
   {
      if(n<0)
         break;
      printf("%lld\n",1+ (n*(n+1)/2));
   }
   return 0;
}
```

```
;UVA PRBLM 10079

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION

   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT

      XOR CX,CX
      XOR BX,BX     ;CLEAR REGISTER
      XOR AX,AX
       XOR DX,DX
       MOV BX,2

       CALL SCAN_NUM  ;SCAN N
       PRINT

       MOV AX,CX
       ADD A,AX

       CMP AX,0
       JE EXIT  ;N<0 BREAK

       ADD AX,1 ;ELSE N+1
```

```
        MUL A     ;N*N+1

        DIV BX         ;/2

        ADD AX,1   ;+1

        PRINTN " "

        CALL PRINT_NUM


          JMP STRT


        EXIT:
        MOV AH,4CH
        INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
Uva 10071:
#include<stdio.h>
int main()
{
   int v,t;
   while(scanf("%d %d",&v,&t)==2)
   {
      if(v==0 && t==0)
      printf("0n");
      else
      printf("%dn",2*v*t);
   }
   return 0;

uva 10019:
;UVA PRBLM 10071

.MODEL SMALL
.STACK 100H
.DATA
V DW 0
T DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT
```

```
DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS



XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

MOV BX,2

 ; PRINTN "ENTER THE VALUE OF V"

CALL SCAN_NUM  ; TAKE THE VALUE OF V AND IT STORES THE VALUE IN CX

printn

 MOV AX,CX
 ADD V,AX  ;V =V+AX

 ;PRINTN "ENTER THE VALUE OF T"

 CALL SCAN_NUM
 printN
 MOV T,CX

  ;COMPARE V AND T WITH 0;IF EQUAL THEN EXIT;OTHERWISE JUMP CALCULATION FOR DISTANCE

 CMP AX,0
 JE EXIT

 JMP CALCULATION

 CMP T,0
 JE EXIT

 JMP CALCULATION

 CALCULATION:
        MUL T
        MUL BX

        ;PRINT " SHORTEST DISTANCE:"

       CALL PRINT_NUm
 EXIT:

    MOV AH,4CH
    INT 21H
    JMP STRT
    RET
;
END MAIN
```

```
//UVA PRBLM 10055
#include<stdio.h>

int main()
{
  long long int c,a,b;

 while( scanf("%lld%lld",&a,&b)==2)
  {
     if(a>b)
    {
  c=a-b;
  }
  else
     c=b-a;
         printf("%lld\n",c);


  }

  return 0;
}
```

```
;#UVA PRBLM 10055

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
  STRT:
  MOV AX,@DATA
  MOV DS,AX     ;DATA INITIALIZATION


  INCLUDE 'EMU8086.INC'
  ;FOR USER INPUT


        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX


         CALL SCAN_NUM  ;DX = A
         MOV DX,CX
         PRINTN
```

```
CALL SCAN_NUM
MOV BX,CX     ;BX = B
PRINTN

CMP DX,BX     ;DX>BX
JG DIFFER      ;JUMP DIFFER
JMP DIFFER1

DIFFER:
    SUB DX,BX    ;DX=DX-BX
    MOV AX,DX    ;AX=DX
    JMP PRINT
DIFFER1:
     SUB BX,DX
     MOV AX,BX     ;BX=BX-DX
                ;AX=BX



PRINT:
   CALL PRINT_NUM   ;PRINT AX


     PRINTN
  JMP STRT
```

```
                EXIT:
                MOV AH,4CH
                INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
```

Uva 10773:

```
      #include<stdio.h>
#include<math.h>
int main()
{
   int n,i;
   double t1,t2,u,v,d;

   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      scanf("%lf%lf%lf",&d,&v,&u);
     if( v==0 || u==0 || v>=u)
      printf("Case %d: can't determine\n", i+1);
     else
      {
            t1 = d/u;

       t2 = (d/sqrt((u*u)-(v*v)));
```

```
    printf("Case %d: %.3lf\n",i+1,t2-t1);
     }
  return 0;}

;UVA PRBLM 10773

.MODEL SMALL
.STACK 100H
.DATA
N DW 0
D DW 0
u DW 0
v DW 0
t1 DW 0
t2 dw 0
i dw 0
U1 DW 0
V1 DW 0
DIFFER DW 0
COUNTER DW 0
D3 DW 0
 B DW 0
 A DW 0
 X DW 0
 ROOT DW 0
.CODE
MAIN PROC

  STRT:
MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
MOV DS,AX

 INCLUDE 'EMU8086.INC'

XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

call SCAN_NUM
mov n,cx
printn

WHILE:
cmp n,1
je exit

call scan_num
;MOV AX,CX      ;D
MOV D,CX
 printn

 call scan_num
 mov DX,cx      ;V
 ADD V,DX
 printn
```

```
call scan_num
mov BX,cx
ADD U,BX       ;U
printn


L1:
 CMP V,0
 JE PRINT1

L2:
CMP U,0
JE PRINT1

L3:
 CMP V,BX
 JGE PRINT1

 T1CAL:
   MOV AX,D
   DIV U
   MOV T1,AX
   XOR AX,AX
   XOR DX,DX



 CALCULATION1:
              ;U*U
```

```
   MOV AX,U
   MUL AX
   MOV U1,AX
   XOR AX,AX


CALCULATION2:
             ;V*V
MOV AX,V
MUL AX
;MOV V1,AX
; XOR AX,AX

DIFFER1:
   SUB U1,AX
   XOR AX,AX
   MOV AX,U1
   MOV D3,AX
   XOR AX,AX

CALL SQRT
MOV BX,X
   MOV ROOT,BX
   MOV AX,D
   DIV ROOT
   MOV T2,AX
   XOR DX,DX
   XOR AX,AX

 MOV CX,T2
```

```
SUB CX,T1
MOV AX,CX
CALL PRINT_NUM

DEC N
PRINTN
JMP WHILE

PRINT1:

PRINTN "CANNOT DETERMINE"
DEC N
JMP WHILE

EXIT:

MOV AH,4CH
INT 21H

MAIN ENDP
```

```
SQRT PROC    ;sqrt function strt

MOV B,2

;CALL SCAN_NUM   ;take a number n
;mov ah,1
; int 21h
MOV AX ,D3      ;ax==n
ADD A,AX
ADD X,AX
mov N, CX       ; x=n
ADD BX,AX       ;bx=n
;DETERMINE SQRT ROOT OF GIVEN NUMBER

DIV B          ;n/2 porjonto loop continue hobe
MOV COUNTER,AX   ;from 1 to n/2
WHILE1:
   CMP COUNTER,1
   JE NEXT      ;(x+(n/x))/2 formula n/2
porjonto continue korle sqrt pabo
   XOR AX,AX
   ADD AX,A     ;ax==n
     xor dx,dx         ;n/x
   DIV X
   ;mov cx,ax              ;x+ax
   ADD Ax,X
   xor dx,dx              ;/2
```

```
    DIV B
    MOV X,ax
    DEC COUNTER
    JMP WHILE1
  NEXT:
  RET
  SQRT ENDP
 DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS
END MAIN
```

```
#uva 11172
#include<stdio.h>
#include <math.h>
int main()
{
  int t,a,b,c,i,d;
  while(scanf("%d %d", &a, &b)==2)
  {
    if(a==0&&b==0)
    break;
    d=0;
    for(i=a;i<=b;i++)
    {
      c=sqrt(i);
      if(c*c==i)
      d++;
    }
    printf("%dn", d);
  }
return 0;}
```

```
;UVA 11172

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
.CODE
MAIN PROC

  MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
  MOV DS,AX

  INCLUDE 'EMU8086.INC'

  ;FOR USER INPUT

  DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS


  XOR CX,CX
  XOR AX,AX
  XOR BX,BX  ;O ASIGN ALL THE RESISTER
```

```
   XOR DX,DX


      PRINTN "ENTER THE VALUE OF A"


   CALL SCAN_NUM  ; TAKE THE VALUE OF V
AND IT STORES THE VALUE IN CX


      MOV AX,CX  ;AX=CX=A



      PRINTN "ENTER THE VALUE OF B"


      CALL SCAN_NUM
    MOV BX,CX    ;BX=CX=B


     ;COMPARE V AND T WITH 0;IF EQUAL THEN
EXIT;OTHERWISE JUMP PRINT FOR PRINT


    CMP AX,BX
    JE PRINT3
     JG PRINT2
     JL PRINT1


     PRINT1:
     PRINTN "<"
     JMP EXIT


     PRINT2:
     PRINTN ">"
      JMP EXIT


    PRINT3:
    PRINTN "="
    JMP EXIT
    EXIT:


       MOV AH,4CH
       INT 21H


END MAIN
```

```
    #uva 10783
//UVA PRBLM 10783
#include<stdio.h>
int main()
{
   int sum,i,a,j,b,n;
   scanf("%d",&n);
   for(i=1;i<=n;i++){
   scanf("%d%d",&a,&b);
    sum =0;
      for(j=a;j<=b;j++)
      {


         if(j%2!=0)
             sum = sum+j;
      }
      printf("Case %d: %d\n",i,sum);
```

```
    }
    return 0;
}

;10783 - Odd Sum

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
I DW 0
J DW 0
SUM1 DW 0
PUT DW 0
COUNT DW 0

.CODE

MAIN PROC
    MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
    MOV DS,AX

    INCLUDE 'EMU8086.INC'

    ;FOR USER INPUT
    XOR AX,AX
    XOR BX,BX
    XOR CX ,CX
    XOR DX,DX

    MOV B,2

    CALL SCAN_NUM   ;INPUT TESTCASE
    MOV DX,CX
    PRINTN
     ADD N,DX
    MOV I,1

    FOR1:
      CMP N,1
      JE EXIT

    CALL SCAN_NUM ; input a
    MOV BX,CX
    ADD PUT,BX

    PRINTN

    CALL SCAN_NUM  ;input b
    MOV CX,CX
    printn

   FOR:

     CMP PUT,CX
     JLE CAL
```

```
   JMP PRINTF

CAL:
    XOR DX,DX
    XOR AX,AX
    ADD AX,PUT
   DIV B
   CMP DX,0
   JNE SUM
    INC PUT
    XOR BX,BX
    ADD BX,PUT
    JMP FOR

SUM:
    ADD COUNT,BX
    INC PUT
    XOR BX,BX
    ADD BX,PUT
    JMP FOR

PRINTF:
    XOR AX,AX
    MOV AX,COUNT
    CALL PRINT_NUM
    DEC N
    JMP FOR1
EXIT:
MAIN ENDP

    DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS
    END MAIN
#uva 10929
#include<stdio.h>
int main()
{
     int a;
    int n;
    while(1)
    {
      scanf("%d",&a);
    if(a==0)
       break;
    else
    {
       if(a%11==0)
          printf("%d is a multiple of 11.\n",a);
       else
          printf("%d is not a multiple of 11.\n",a);
    }

    }
    return 0;
}
```

```
;UVA PRBLM 10929

.MODEL SMALL
.STACK 100H
.DATA
M DW 0
N DW 0
A DW 0
.CODE
MAIN PROC

  STRT:
  MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
  MOV DS,AX

  INCLUDE 'EMU8086.INC'

  ;FOR USER INPUT

  DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS


  XOR CX,CX
  XOR AX,AX
  XOR BX,BX  ;O ASIGN ALL THE RESISTER
  XOR DX,DX
  MOV N,11

  CALL SCAN_NUM
  ADD A,CX
  MOV M,CX

  PRINTN
  CMP M,0
  JE EXIT
  JMP CALCULATION

  CALCULATION:
      MOV AX,M
       DIV N
       CMP DX,0
       JE PRINT1
       JNE PRINT2

  PRINT1:
    XOR DX,DX
    XOR AX,AX
    MOV AX,A
   CALL PRINT_NUM
   PRINTN "IS A MULTIPLE OF 11."

   JMP STRT

  PRINT2:
```

```
   XOR DX,DX
   XOR AX,AX
    MOV AX,A
    CALL PRINT_NUM
    PRINTN " :IS NOT A MULTIPLE OF 11."


          JMP STRT

   EXIT:

      MOV AH,4CH
      INT 21H

;
END MAIN
#uva 11854
//UVA PRBLM 11854
#include<stdio.h>
#include<math.h>
int main()
{
   int a,b,c,d,i,n;
   while(scanf("%d%d%d",&a,&b,&c)==2)
   {
      if(a==0&&b==0&&c==0)
         break;
      else
      {
         if(c==sqrt((a*a)+(b*b)))
            printf("right\n");
         else
            printf("wrong\n");

      }
   }
   return 0;
}
;UVA PRBLM 11854

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
C DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT

   DEFINE_SCAN_NUM
```

```
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS



XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

; PRINTN "ENTER A"

CALL SCAN_NUM

MOV AX,CX
ADD A,AX
IMUL A
MOV A,AX
CMP A,0  ;A==0 THEN EXIT
JE EXIT
XOR AX,AX

;PRINTN "ENTER B"
CALL SCAN_NUM
MOV AX,CX
ADD B,AX
IMUL B
MOV BX,AX

CMP BX,0
JE EXIT
XOR AX,AX


ADD BX,A


;PRINTN "ENTER C"
CALL SCAN_NUM
MOV AX,CX
ADD C,AX
IMUL C
MOV DX,AX

CMP DX,0
JE EXIT
XOR AX,AX


CMP BX,DX ;C2==A2 *B2
JE PRINTR

PRINTN "WRONG"
JMP EXIT

PRINTR:
PRINTN "RIGHT"
 EXIT:
```

```
MOV AH,4CH
        INT 21H

END MAIN
//UVA PRBLM 10300

#include<stdio.h>
int main()
{
long long a,t,n,b,c,i,j,ans;
while(scanf("%lld",&t)==1)
   {
   for(i=0;i<t;i++)
      {
      ans=0;
      scanf("%lld",&n);
      for(j=0;j<n;j++)
         {
         scanf("%lld%lld%lld",&a,&b,&c);
         ans+=a*c;
         }
      printf("%lld\n",ans);
      }
   }
return 0;
}
;UVA PRBLM 10300

.MODEL SMALL
.STACK 100H
.DATA
V DW 0
T DW 0
ANS DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS



   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

   CALL SCAN_NUM
   MOV V,CX
```

```
PRINTN
COMP:
CMP V,0
JE PRINT
JMP CALCULATION
  CALCULATION:

  ;PRINTN "ENTER THE VALUE OF A"

CALL SCAN_NUM  ; TAKE THE VALUE OF V
AND IT STORES THE VALUE IN CX

  MOV AX,CX
  PRINTN



; PRINTN "ENTER THE VALUE OF B"

 CALL SCAN_NUM
MOV BX,CX
PRINTN
; PRINTN "ENTER THE VALUE OF C"

CALL SCAN_NUM
MOV DX,CX
PRINTN



IMUL DX ;AX=AX*DX=A*C
ADD ANS,AX
DEC V
JMP COMP

PRINT:

   MOV AX,ANS
   CALL PRINT_NUM
   JMP EXIT

 JMP STRT
EXIT:

   MOV AH,4CH
   INT 21H
   JMP STRT
   RET
;
END MAIN
//UVA PRBLM 12577

#include<stdio.h>
#include<string.h>
int main(){
char ary[6];
int counter=0;
while(1)
{
  gets(ary);
```

```
   if(ary[0]=='*')
      {
         break;
      }
   else if(ary[0]=='H')
      {
         printf("Case %d: Hajj-e-
Akbar\n",++counter);
      }
   if(ary[0]=='U')
      {
         printf("Case %d: Hajj-e-
Asghar\n",++counter);
      }
}
return 0;
}
```

```
;12577 - Hajj-e-Akbar

ORG 100H

.DATA
Hajj DB 'Hajj-e-Akbar', 10, 13, '$'  ;10,13 NEW
LINE
Umrah DB 'Hajj-e-Asghar', 10, 13, '$'

.CODE

MAIN PROC
   START:
   MOV AH, 1   ;INPUT
   INT 21H
   MOV BL, AL   ;BL=AL
   HUDDAI:
   INT 21H    ;PRESS ENTER
   CMP AL, 13
   JNE HUDDAI

   CMP BL, '*'
   JE ENDMAIN
   CMP BL, 'H'
   JE AKBAR

   LEA DX, UMRAH
   MOV AH, 9
   INT 21H
   JMP START

   AKBAR:
   LEA DX, Hajj
   MOV AH, 9
   INT 21H
   JMP START

   ENDMAIN:
   RET
```

```
#uva 12802
#include<stdio.h>
long long palindrm(int n)
```

```
{
   int t,r;
   r=0;

   t=n;
   while(t!=0)
   {
      r=r*10;
      r=r+t%10;
      t=t/10;
   }
   if(n==r)
      return 1;
   return 0;
}
;UVA 12802

 INCLUDE 'EMU8086.INC'
.MODEL SMALL
.STACK 100H
.DATA
 A DW 0
 N DW 0
 B DW 0
 D DW 2
VAR1 DW ?
TEMP DW ?

.CODE
```

```
MAIN PROC

    STRT:

    XOR AX,AX
    XOR BX,BX
    XOR CX,CX
    XOR DX,DX
    MOV D,2

  CALL SCAN_NUM
  MOV  AX,CX
  PRINTN

  ADD N,AX

   MUL D
   MOV AX,AX
  CALL PRINT_NUM
  PRINTN


 CALL PALLINDROM
 MOV AX,A

  CALL PRIME
```

```
 MOV BX,B

CHECK1:
CMP AX,1
JE CHECK2
JMP EXIT

CHECK2:
   CMP BX,1
   JE STRT
   JMP EXIT

MAIN ENDP
;FOR USER INPUT
DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS

PALLINDROM PROC

AND BX,0      ;BX=0
AND CX,0      ;CX=0
MOV CL,10D    ;CL=10

NUMBER_SAVE:
MOV VAR1,BX

MOV AX,BX
XOR BX,BX
XOR DX,DX

CHECK:
DIV CL
;MOV TEMP,AL
MOV DL,AH
XOR AH,AH
MOV TEMP,AX
MOV AX,BX
MUL CL
ADD AX,DX
MOV BX,AX
MOV AX,TEMP
CMP AX,0
JE BREAK
JMP CHECK

BREAK:
```

```
  MOV DX,VAR1

  CMP BX,DX
  JE PRINT_SAME
  JMP PRINT_DIFF

  PRINT_SAME:
  PRINTN
  XOR AX,AX
   MOV A,1
   ;MOV AX,A
    JMP EXIT


  PRINT_DIFF:
  PRINTN
  XOR AX,AX
  MOV A,0
  ;MOV AX,A
  EXIT:
  RET
PALLINDROM ENDP


PRIME PROC
  mov ax,@data
  mov ds,ax    ;data initialization
   call scan_num    ;scan num from keyboard & it stores the value in cx

   mov ax,cx

   mov b,2
   div b

    mov a,0

   cmp ax,a
   je print1
    MOV B,0
    JMP EXIT1


   print1:
    MOV B,1


   exit1:
   RET
    mov ah,4ch
```

```
   int 21h
end PRIME

END MAIN
```

```
;uva 11364
#include<stdio.h>
#include<string.h>
int main()
{
  int t,d,c,n;
  scanf("%d",&t);
  while(t--)
  {
     scanf("%d",&c);
     int mx=0,mn=100;
     while(c--)
     {
        scanf("%d",&n);
        if(n>mx)
         mx=n;
        if(n<mn)
         mn=n;
      }
      d = 2*(mx-mn);
      printf("%d\n",d);

        }
   return 0;
}
```

```
;uva 11364
.MODEL SMALL
.STACK 100H
.DATA
A DW 0
MX DW 0
MN DW 100
N DW 0
.CODE
MAIN PROC

   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT
      XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

   ;PRINT "N:"

   CALL SCAN_NUM
   MOV A,CX
   XOR CX,CX
   PRINTN
```

```
  LOOP1:
CMP A,0
JE CALCULATION

 ;PRINT "NUM: "
 XOR AX,AX
CALL SCAN_NUM
 PRINTN     ;;N==AX
MOV AX,CX
ADD N,AX
XOR CX,CX


 CMP AX,MX
 JG L1


L1:
  MOV MX,AX


  XOR AX,AX
 MOV AX,N
 CMP AX,MN
 JL L2
```

```
L2:

  MOV MN,AX


  DEC A

  JMP LOOP1

CALCULATION:
        XOR CX,CX
        XOR AX,AX
        XOR BX,BX
        XOR DX,DX


        MOV CX,2
        ADD BX,MX
        ADD DX,MN
        SUB BX,DX
        MOV AX,BX
        IMUL CX
        MOV AX,CX


CALL PRINT_NUM
JMP EXIT


 EXIT:
```

```
    MOV AH,4CH
    INT 21H


 MAIN ENDP

 DEFINE_SCAN_NUM
 DEFINE_PRINT_NUM
 DEFINE_PRINT_NUM_UNS

END MAIN
//UVA PRBLM 12372
#include<stdio.h>
int main()
{
  int t,l,w,h,count = 0;
  scanf("%d",&t);
  while(t--)
  {
    scanf("%d%d%d",&l,&w,&h);
    if(l<=20 && w<=20 && h<=20)
      printf("Case %d: good\n",++count);
    else
      printf("Case %d: bad\n",++count);
  }
  return 0;
}
;UVA PRBLM 12372
```

```
.MODEL SMALL
.STACK 100H
.DATA
V DW 0
T DW 0
.CODE
MAIN PROC

  MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
  MOV DS,AX

  INCLUDE 'EMU8086.INC'

  ;FOR USER INPUT

  DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS


  XOR CX,CX
  XOR AX,AX
  XOR BX,BX  ;O ASIGN ALL THE RESISTER
  XOR DX,DX

 CALL SCAN_NUM
```

```
MOV AX,CX
 PRINTN      ;SCAN L
; XOR CX,CX

CALL SCAN_NUM

MOV BX,CX      ;SCAN W
 PRINTN
;XOR CX,CX

CALL SCAN_NUM     ;SCAN H
PRINTN
MOV  DX,CX

CMP AX,20     ;IF CONDITION
JLE  LEVEL1
JMP PRINT2

LEVEL1:
    CMP BX,20
    JLE LEVEL2
    JMP PRINT2

LEVEL2:
     CMP DX,20
     JLE PRINT1
     JMP PRINT2

PRINT1:
     PRINTN " CASE 1: GOOD "
     JMP EXIT

PRINT2:
     PRINTN " CASE 1: BAD "


  EXIT:

      MOV AH,4CH
      INT 21H

      RET
;
END MAIN
   #uva 11479
 #include<stdio.h>
int main()
{
long int t,a,b,c,i;
while(scanf("%ld",&t)==1)
{
i=1;
while(i<=t)
 {
 scanf("%ld%ld%ld",&a,&b,&c);
 if((a+b)<=c || (b+c)<=a || (c+a)<=b)
```

```
 printf("Case %ld: Invalid\n",i);
 else if(a<=0 || b<=0 || c<=0)
 printf("Case %ld: Invalid\n",i);
 else if(a==b && b==c)
 printf("Case %ld: Equilateral\n",i);
 else if(a==b || b==c || c==a)
 printf("Case %ld: Isosceles\n",i);
 else
 printf("Case %ld: Scalene\n",i);
 i++;
 }

}
return 0;
}
;#UVA PRBLM 11479

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
C DW 0
SUM1 DW 0
SUM2 DW 0
SUM3 DW 0
N DW 0
M DW 0
P DW 0
```

```
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT


      XOR CX,CX
      XOR BX,BX   ;CLEAR REGISTER
      XOR AX,AX
      xor dx,dx

      CALL SCAN_NUM
      MOV AX,CX      ;INPUT A
      ADD A,AX
      ADD N,AX
      PRINTN


      CALL SCAN_NUM
      MOV BX,CX
      ADD B,BX
      ADD M,BX    ;INPUT B
```

```
PRINTN

CALL SCAN_NUM
MOV CX,CX
ADD C,CX
ADD P,CX      ;INPUT C
PRINTN


S:
XOR AX,AX
MOV AX,A
ADD AX,B ;A+B
MOV SUM1,AX

 S1:
 XOR AX,AX

MOV AX,B
ADD AX,C  ;B+C
MOV SUM2,AX

 S3:
 XOR AX,AX
 MOV AX,C
ADD AX,N  ;C+A
MOV SUM3,AX

L1:
 XOR AX,AX
 MOV AX,SUM1
 CMP AX,P
 JLE PRINT1
L2:
  XOR AX,AX
  MOV AX,SUM2
  CMP AX,M
 JLE PRINT1

L3:
  XOR AX,AX
  MOV AX,SUM3
 CMP AX,N
 JLE PRINT1

L4:
  CMP M,0
  JLE PRINT1
 L5:
 CMP N,0
 JLE PRINT1
 L6:
 CMP P,0
 JLE PRINT1

CHECK:
  XOR AX,AX
  ADD AX,M  ;A==B
```

```
  CMP AX,N
  JE CHECK1

CHECK1:
    XOR AX,AX
    ADD AX,N    ;B==C
    CMP AX,P
     JE PRINT2

     CHECK12:
  XOR AX,AX
  ADD AX,M   ;A==B
  CMP AX,N
  JE PRINT3

   CHECK13:
    XOR AX,AX
    ADD AX,N    ;B==C
    CMP AX,P
     JE PRINT3


CHECK2:
    XOR AX,AX
    ADD AX,P
    CMP AX,M
    JE PRINT3


PRINT1:
    PRINTN " INVALID"
    JMP EXIT

PRINT2:
    PRINTN "EQUILATERAL"
    JMP EXIT

PRINT3:
    PRINTN "ISOSCELES"
     JMP EXIT

   PRINTN "SCALENE"


    EXIT:
    MOV AH,4CH
    INT 21H


  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
//UVA PRBLM 10079
```

```
#include<stdio.h>
int main()
{
   int p;
   long long int n;
   while(scanf("%lld",&n))
   {
      if(n<0)
         break;
      printf("%lld\n",1+ (n*(n+1)/2));
   }
   return 0;
}
```

```
;UVA PRBLM 10079

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION
```

```
INCLUDE 'EMU8086.INC'
;FOR USER INPUT

        XOR CX,CX
        XOR BX,BX      ;CLEAR REGISTER
        XOR AX,AX
        XOR DX,DX
        MOV BX,2

        CALL SCAN_NUM  ;SCAN N
        PRINT

        MOV AX,CX
        ADD A,AX

        CMP AX,0
        JE EXIT  ;N<0 BREAK

        ADD AX,1 ;ELSE N+1

        MUL A     ;N*N+1

        DIV BX         ;/2
```

```
        ADD AX,1   ;+1

        PRINTN " "

        CALL PRINT_NUM


          JMP STRT



        EXIT:
        MOV AH,4CH
        INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //UVA PRBLM 11498
#include<stdio.h>
int main()
{
   int n,m,i,x,y,k;
   while(scanf("%d",&k)==1)
   {
      if(k==0)
         break;
      scanf("%d%d",&n,&m);
      for(i=0; i<k; i++)
      {
         scanf("%d%d",&x,&y);
         if(n==x || m==y)
            printf("divisa\n");
         else if(x>n && y>m)
            printf("NE\n");
         else if(x<n&&y>m)
         {
            printf("NO\n");
         }
         else if(x<n&&y<m)
         {
            printf("SO\n");
         }
         else if(x>n&&y<m)
         {
            printf("SE\n");
         }

      }
   }
   return 0;
}
;#UVA PRBLM 11498

.MODEL SMALL
```

```
.STACK 100H
.DATA
A DW 0
N DW 0
M DW 0
I DW 0
K DW 0
X DW 0
Y DW 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT



        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX


         CALL SCAN_NUM
MOV K,CX
PRINTN        ;CX = K

CMP K,0
JE EXIT

CALL SCAN_NUM
MOV N,CX
PRINTN

CALL SCAN_NUM
MOV M,CX
PRINTN

MOV DX,0  ;I=0=DX

LOOP1:
   CMP DX,K
   JG EXIT

   CALL SCAN_NUM
   MOV AX,CX      ;X=AX
   PRINTN

   CALL SCAN_NUM
   MOV BX,CX      ;Y=BX

CMP AX,N
 JE CHECK1   ;X== N
```

```
 CMP AX,N
 JG CHECK2   ;X>N

 CMP AX,N   ;X<N
 JL CHECK3

 CMP AX,N
 JL CHECK4  ;X<N

CHECK1:
    CMP BX,M  ;Y==M
    JE PRINT1

CHECK2:
     CMP BX,M
     JG PRINT2  ;Y>M


CHECK3:
      CMP BX,M
      JG PRINT3
CHECK4:
    CMP BX,M
    JL PRINT4

PRINT1:
    PRINTN "DIVISA"
    INC DX
   JMP LOOP1

PRINT2:
     PRINTN "NE"
     INC DX
    JMP LOOP1

PRINT3:
     PRINTN "NO"
     INC DX
     JMP LOOP1

PRINT4:
      PRINTN "SO"
      INC DX

    JMP LOOP1


     EXIT:
     MOV AH,4CH
     INT 21H
MAIN ENDP
```

```
 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //UVA PRBLM 12646
#include <stdio.h>
int main()
{
int a,b,c;
while(scanf("%d %d %d", &a, &b, &c) == 3)
{
   if(a == b && b == c)
     printf("*\n");
   else if(a != b && b == c)
         printf("A\n");
   else if(a != b && a == c)
         printf("B\n");
   else if(a == b && a != c)
         printf("C\n");
}
return 0;
}
;UVA PRBLM 12646

.MODEL SMALL
.STACK 100H
.DATA
V DW 0
T DW 0
.CODE
MAIN PROC

   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT


   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

   CALL SCAN_NUM
            ;SCAN A
   MOV AX,CX
    PRINTN
   ; XOR CX,CX

   CALL SCAN_NUM
             ;SCAN B
   MOV BX,CX
    PRINTN
   ;XOR CX,CX
```

```
 CALL SCAN_NUM     ;SCAN C
 PRINTN
MOV  DX,CX

LEVEL1:
    CMP AX,BX
    JE LEVEL3    ;AX==BX
    JNE LEVEL8

LEVEL2:
    CMP AX,DX    ;AX==DX
    JE LEVEL4



LEVEL3:
     CMP BX,DX  ;BX==DX
     JE PRINT1

LEVEL4:
     CMP AX,BX
     JNE PRINT3 ;AX!=BX

LEVEL5:
    CMP AX,DX     ;AX!=DX
    JNE LEVEL7

LEVEL6:
    CMP BX,DX
    JE PRINT2  ;BX==DX
LEVEL7:
    CMP AX,BX
    JE PRINT4   ;AX==BX
LEVEL8:
    CMP AX,DX    ;AX==DX
     JE PRINT3


 PRINT1:
      PRINTN "*"
      JMP EXIT

 PRINT2:
     PRINTN "A"
     JMP EXIT

PRINT3:
     PRINTN "B"
     JMP EXIT

 PRINT4:
      PRINTN "C"
      JMP EXIT



 EXIT:

    MOV AH,4CH
```

```
        INT 21H



    MAIN ENDP

    DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS



;
END MAIN
```

```
    //uva 12952
#include<stdio.h>
int main()
{
    int a,b;
    while( scanf("%d%d",&a,&b)!=EOF)
    {
        if(a>=b)
            printf("%d\n",a);
        else
            printf("%d\n",b);
    }
    return 0;
}
```

```
;uva 12952
;#UVA PRBLM 12952

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
    STRT:
    MOV AX,@DATA
    MOV DS,AX     ;DATA INITIALIZATION



    INCLUDE 'EMU8086.INC'
    ;FOR USER INPUT



        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
        XOR DX,DX
```

```
    CALL SCAN_NUM  ;DX = A
    MOV DX,CX
    PRINTN

    CALL SCAN_NUM
    MOV BX,CX     ;BX = B
    PRINTN

   CMP DX,BX
   JGE L1

   L2:
     MOV AX,BX
     CALL PRINT_NUM
     JMP EXIT
   L1:
     MOV AX,DX
     CALL PRINT_NUM



      EXIT:
      MOV AH,4CH
      INT 21H

MAIN ENDP
 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //UVA PRBLM 10696
#include<stdio.h>
int main()
{
   int n,N=0;
    while(scanf("%d",&n)==1&&n!=0)
   {
      if(n <= 101)
      printf("f91(%d) = 91\n",n);
      else
      printf("f91(%d) = %d\n",n,n-10);
   }
   return 0;
}
//;UVA PRBLM 10696

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
C DW 0
.CODE
MAIN PROC
```

```
MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
MOV DS,AX

INCLUDE 'EMU8086.INC'

;FOR USER INPUT

DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS



XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

call SCAN_NUM ;SCAN N
MOV AX,CX
PRINTN

CMP AX,101      ;CHECK IF CONDITION
JLE PRINT
JMP CALCULATION

PRINT:
PRINT "F91"
PRINT "("

CALL PRINT_NUM
PRINT ")"
PRINT "= 91"
PRINTN " "
JMP EXIT

CALCULATION:
        ;ADD A,AX
       ; PUSH AX
        MOV BX,AX
       SUB BX,10
       MOV AX,BX
       PRINT "F91 ="
       PRINT "("
       CALL PRINT_NUM
       PRINT ")"
       ;POP CX
       PRINTN " "
       ;POP BX

  EXIT:

    MOV AH,4CH
    INT 21H

;
END MAIN
```

```
   //UVA PRBLM 11150
#include<stdio.h>
int main()
{
int a,n;
while(scanf("%d",&n)==1)
{
a=n;
while(n>=3)
{
a=a+(n/3);
n=(n/3)+(n%3);
}
if(n==2)
{
a++;
}
printf("%d\n",a);
}
return 0;
}
```

```
;UVA PRBLM 11150
.MODEL SMALL
.STACK 100H
.DATA
A DW 0
T DW 0
.CODE
MAIN PROC
    MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
    MOV DS,AX

    INCLUDE 'EMU8086.INC'

    ;FOR USER INPUT


    XOR CX,CX
    XOR AX,AX
    XOR BX,BX  ;O ASIGN ALL THE RESISTER

   CALL SCAN_NUM      ;N==AX

   MOV AX,CX
   XOR CX,CX
    PRINTN
    MOV A,AX

    MOV BX,3     ;BX ==3

    WHILE:
        CMP AX,BX
        JGE CALCULATION
        JMP IF
```

```
CALCULATION:
        XOR DX, DX
        DIV BX
        ADD A,AX
        ADD AX,DX
       JMP WHILE
IF:
 CMP AX,2
 JE LOOP1
 JMP PRINT

LOOP1:
 INC A
 JMP PRINT

PRINT:
  MOV AX,A
  CALL PRINT_NUM



 EXIT:

    MOV AH,4CH
    INT 21H
```

```
    MAIN ENDP

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS

END MAIN
```

```
//uva    12917
    #include<stdio.h>
int main()
{
   int x,y,z;
   while(scanf("%d%d%d",&x,&y,&z)==3)
   {

      if(x<=(z-y))
         printf("Props win!\n");
      else
         printf("Hunters win!\n");
   }
   return 0;
}
;#UVA PRBLM 12917

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
```

```
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT



       XOR CX,CX
       XOR BX,BX   ;CLEAR REGISTER
       XOR AX,AX
        XOR DX,DX

       CALL SCAN_NUM
       MOV DX,CX      ;X INPUT
       PRINTN

       CALL SCAN_NUM
       MOV BX,CX      ;Y INPUT
       PRINTN

       CALL SCAN_NUM
       MOV AX,CX       ;Z INPUT
       PRINTN

       SUB AX,BX
       MOV A,AX    ;Z-Y

       CMP DX,A
       JLE PRINT12

       PRINTN "Hunters win!"
       jmp strt

       PRINT12:
          printn "Props win!"

          jmp strt


          EXIT:
          MOV AH,4CH
          INT 21H
    MAIN ENDP

   DEFINE_SCAN_NUM
     DEFINE_PRINT_NUM
     DEFINE_PRINT_NUM_UNS
  END MAIN
```

```
//UVA PRBLM 12992
#include<stdio.h>
int main()
{
   int n,m,count = 1;
   scanf("%d",&n);
   while(n--)
   {

      scanf("%d",&m);
       printf("Case #%d: %d\n",count++,2*m-1);
   }
   return 0;
}
```

```
;#UVA PRBLM  12992

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
I DW 0
J DW 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION
```

```
INCLUDE 'EMU8086.INC'
;FOR USER INPUT


      XOR CX,CX
      XOR BX,BX   ;CLEAR REGISTER
      XOR AX,AX
       XOR DX,DX
       MOV B,2
 CALL SCAN_NUM
 MOV BX,CX
 mov a,0
 PRINTN

 WHILE:
    CMP BX,A
    JE EXIT

    CALL SCAN_NUM
    MOV AX,CX
    PRINTN

    MUL B
    SUB AX,1
    MOV DX,AX
```

```
        INC A

       PRINT "Case #"
        MOV AX,A
        CALL PRINT_NUM
        PRINT ":"
        MOV AX,DX
        CALL PRINT_NUM
        PRINTN

        JMP WHILE



            EXIT:
            MOV AH,4CH
            INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //UVA PRBLM 13012
#include<stdio.h>
int main()
{
   int n,m,i,count;
   while(scanf("%d",&n)!=EOF)
   {
      count =0;


         for(i=0;i<5;i++){
         scanf("%d",&m);
         if(n==m)
         count++;

         }


      printf("%d\n",count);
   }
   return 0;
}
;#UVA PRBLM  13012

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
I DW 0
J DW 0
.CODE
MAIN PROC
```

```
STRT:
MOV AX,@DATA
MOV DS,AX     ;DATA INITIALIZATION


INCLUDE 'EMU8086.INC'
;FOR USER INPUT


    XOR CX,CX
    XOR BX,BX   ;CLEAR REGISTER
    XOR AX,AX
     XOR DX,DX

  call scan_num
  MOV BX,CX
  PRINTN
  MOV A,0


  CAL:
    CMP BX,A
    JE PRINT1
     INC A
    CALL SCAN_NUM
    MOV DX,CX
    PRINTN
    CMP DX,BX
     JE LOOP1
     JMP CAL


LOOP1:
    INC J
    JMP CAL


    PRINT1:
       MOV AX,J
       ;PRINTN
       CALL PRINT_NUM
       PRINTN


       EXIT:
       MOV AH,4CH
       INT 21H

   MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
//uva 13025
      #include<stdio.h>
```

```
int main()
{
    printf("May 29, 2013 Wednesday\n");
   return 0;}
;#UVA PRBLM  13025

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION



   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT




        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
                XOR AX,AX
                 XOR DX,DX

                PRINTN " May 29,2013 Wednesday"



                    EXIT:
                    MOV AH,4CH
                    INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //uva 12149
#include<stdio.h>
int main()
{
   int n,s;
   while(scanf("%d",&n)==1)
   {
      if(n==0)
         break;
         s=(n*(n+1)*(2*n+1))/6;
      printf("%d\n",s);
   }
```

```
   return 0;
}
;UVA PRBLM 12149

.MODEL SMALL
.STACK 100H
.DATA
N DW 0
N1 DW 0
TEMP DW 0
N2 DW 0
A DW 0
B DW 0
I DW 0

.CODE
MAIN PROC

   STRT:

   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS

   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

   MOV A,2
   MOV B,6
   CALL SCAN_NUM
   MOV AX,CX
   ADD N,AX
   ADD N1,AX
   PRINTN

   CMP AX,0
   JE EXIT

    FIRSTCAL:   ;N*(N+1)
   ADD N,1
   MUL N
    mov bx,AX

   SECONDCAL:
       MOV AX,N1
       MUL A      ;N*2+1
```

```
        ADD AX,1



    THIRDCAL:
        MUL BX



    FINALCAL:
        DIV B
        CALL PRINT_NUM
        PRINTN
        XOR AX,AX

        JMP STRT



    EXIT:

       MOV AH,4CH
       INT 21H

END MAIN



    //UVA PRBLM 10970
#include<stdio.h>
int main()
{
int m,n,c;
while(scanf("%d%d",&m,&n)==2)
printf("%d\n",m*n-1);
return 0;}
;UVA PRBLM 10970


.MODEL SMALL
.STACK 100H
.DATA
M DW 0
N DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS



   XOR CX,CX
```

```
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX



 PRINTN "ENTER THE VALUE OF M"

CALL SCAN_NUM  ; TAKE THE VALUE OF V
AND IT STORES THE VALUE IN CX

 MOV AX,CX    ;AX=CX=M


 PRINTN "ENTER THE VALUE OF N "

 CALL SCAN_NUM
 MOV BX,CX          ;BX=CX=N
 SUB BX,1           ;CX= N-1=BX-1


 IMUL BX

 PRINTN " BIG CHOLCATE PICES:"
 CALL PRINT_NUM



        JMP STRT

 EXIT:

    MOV AH,4CH
    INT 21H

;
END MAIN
```

```
 //uva 10110
#include<stdio.h>
#include<math.h>
int main()
{
  long long int a,b,c,d;

  while(scanf("%lld",&a)==1)
  {
    if(a==0)
      break;
    b = sqrt(a);
    c = b*b;
    if(c==a)
      printf("yes\n");
    else
```

```
        printf("no\n");
    }
    return 0;
}
;UVA PRBLM 10110

.MODEL SMALL
.STACK 100H
.DATA
N DW 0
X DW 0
A DW 0
B DW 0
C DW 0
COUNTER DW 0
.CODE
MAIN PROC

    STRT:
    MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
    MOV DS,AX

     INCLUDE 'EMU8086.INC'


    XOR CX,CX
    XOR AX,AX
    XOR BX,BX  ;O ASIGN ALL THE RESISTER
    XOR DX,DX
```

```
CALL SQRT   ;call sqrt function

 MOV AX,BX
 MUL AX
 ; MOV C,AX

 CMP N,AX
 JE PRINT1

 PRINTN "NO"
JMP EXIT

PRINT1:
   PRINTN "YES"

EXIT:

   MOV AH,4CH
   INT 21H

   MAIN ENDP


SQRT PROC     ;sqrt function strt

   MOV B,2d
```

```
CALL SCAN_NUM   ;take a number n
;mov ah,1
; int 21h
MOV AX ,CX      ;ax==n
ADD A,AX
ADD X,AX
mov N, CX       ; x=n
ADD BX,AX       ;bx=n
;DETERMINE SQRT ROOT OF GIVEN NUMBER

DIV B           ;n/2 porjonto loop continue hobe
MOV COUNTER,AX  ;from 1 to n/2
WHILE:
    CMP COUNTER,1
    JE NEXT     ;(x+(n/x))/2 formula n/2 porjonto continue korle sqrt pabo
    XOR AX,AX
    ADD AX,A    ;ax==n
      xor dx,dx         ;n/x
    DIV BX
    mov cx,ax           ;x+ax
    ADD ax,BX
    xor dx,dx           ;/2
    DIV B
    MOV BX,ax
    DEC COUNTER
    JMP WHILE
  NEXT:
    RET
    SQRT ENDP

    ;FOR USER INPUT

    DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS

;
END MAIN
```

```
    //UVA PRBLM 10346
#include<stdio.h>
int main()
{
int a,n,k;
while(scanf("%d %d",&n,&k)==2 && k>1)
{
a=n;
while(n>=k)
{
a=a+(n/k);
n=(n/k)+(n%k);
}
printf("%d\n",a);
}
return 0;
}
```

```
;UVA PRBLM 10346

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC

   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION



   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS

        ;XOR B,B
        ;XOR R,R
       XOR CX,CX
       XOR BX,BX
       XOR AX,AX
       XOR DX,DX

       PRINTN "ENTER VALUE N:"

   CALL SCAN_NUM  ;IT STORES THE VALUE IN CX


   MOV ax,CX  ;SCANF THE VALUE OF N
   ADD N,AX
   ADD A,AX
     PRINTN "ENTER VALUE K:"



   CALL SCAN_NUM
   MOV CX,CX   ;SCANF THE VALUE OF K


   WHILE:
   CMP N,CX
   JGE LOOP1
  JMP POPOUT

   LOOP1:
      DIV CX
      MOV B,AX

      MOV R,DX

      ADD AX,B
```

```
        ADD DX,B
        MOV N,DX

        JMP WHILE


   POPOUT:
       ;
        ;MOV AH,2
        ;ADD AX,48
        ;MOV DX,AX
         call print_num
        ;INT 21H

  ; MAIN ENDP
END MAIN
//UVA PRVLM 11172

#include<stdio.h>
int main()
{
    int n,a,b;
    scanf("%d",&n);

        while(scanf("%d%d",&a,&b)!=n){
        if(a>b)
    {
        printf(">\n");
    }
    else if(a<b)
        printf("<\n");
        else
            printf("=\n");}
        return 0;

}
;UVA 11172

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
.CODE
MAIN PROC

    MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
    MOV DS,AX

    INCLUDE 'EMU8086.INC'

    ;FOR USER INPUT

    DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS
```

```
   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

     PRINTN "ENTER THE VALUE OF A"

   CALL SCAN_NUM  ; TAKE THE VALUE OF V
AND IT STORES THE VALUE IN CX

     MOV AX,CX  ;AX=CX=A

     PRINTN "ENTER THE VALUE OF B"

     CALL SCAN_NUM
    MOV BX,CX    ;BX=CX=B

    ;COMPARE V AND T WITH 0;IF EQUAL THEN
EXIT;OTHERWISE JUMP PRINT FOR PRINT

    CMP AX,BX
    JE PRINT3
     JG PRINT2
     JL PRINT1

     PRINT1:
     PRINTN "<"
     JMP EXIT

     PRINT2:
     PRINTN ">"
     JMP EXIT

    PRINT3:
    PRINTN "="
    JMP EXIT
    EXIT:

       MOV AH,4CH
       INT 21H

END MAIN
//uva 11461
#include<stdio.h>
#include<math.h>
int main()
{
   int a,b,n,c,i;

      while(scanf("%d%d",&a,&b)==2){
      int d=0;
      if(a==0&&b==0)
         break;
```

```
    else
    {


    for(i=a;i<=b;i++)
    {
       c=sqrt(i);
       if(c*c==i)
          d++;


    }}

     printf("%d\n",d);}


  return 0;
}
```

```
;#UVA PRBLM 11461

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
X DW 0
C DW 0
COUNT1 DW 0
COUNT2 DW 0
I DW 0
D DW 0
COUNTER DW 0

.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT
```

```
        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
        XOR DX,DX

        CALL SCAN_NUM
        MOV BX,CX
        ADD COUNT1,BX   ;A
        PRINTN

        CALL SCAN_NUM
        MOV DX,CX
```

```
ADD COUNT2,DX     ;B
PRINTN

CMP BX,0
JE EXIT

CMP DX,0
JE EXIT

MOV I,BX  ;I==A

FORLOOP:

    CMP I,DX
    JGE PRINT1

    CALL SQRT
    MOV Cx,bX
   ; INC I

    MOV AX,Cx
    MUL AX

    CMP AX,I
    JE L1
    INC I
    JMP FORLOOP

 L1:
    INC D
    INC I
    JMP FORLOOP

PRINT1:
   MOV AX,D
    CALL PRINT_NUM
    PRINTN

    EXIT:
    MOV AH,4CH
    INT 21H

  MAIN ENDP

SQRT PROC     ;sqrt function strt

    MOV B,2

  ; CALL SCAN_NUM   ;take a number n
   ;mov ah,1
  ; int 21h
  XOR AX,AX
   MOV AX ,I       ;ax==n
   ADD A,AX
   ADD X,AX
   mov N, CX        ; x=n
```

```
    ADD BX,AX         ;bx=n
    ;DETERMINE SQRT ROOT OF GIVEN NUMBER

    DIV B           ;n/2 porjonto loop continue hobe
    MOV COUNTER,AX    ;from 1 to n/2
    XOR DX,DX
    WHILE:
        CMP COUNTER,1
        JE NEXT       ;(x+(n/x))/2 formula n/2
porjonto continue korle sqrt pabo
      XOR AX,AX
      ADD AX,A      ;ax==n
          xor dx,dx          ;n/x
      DIV bX
      mov cx,ax                ;x+ax
      ADD ax,bX
      xor dx,dx                ;/2
      DIV B
      MOV bX,ax
      DEC COUNTER
      JMP WHILE
     NEXT:
     RET
     SQRT ENDP



 DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS
END MAIN
```

```
      //UVA PRBLM 10783
#include<stdio.h>
int main()
{
    int sum,i,a,j,b,n;
    scanf("%d",&n);
    for(i=1;i<=n;i++){
    scanf("%d%d",&a,&b);
     sum =0;
       for(j=a;j<=b;j++)
       {

          if(j%2!=0)
              sum = sum+j;
       }
       printf("Case %d: %d\n",i,sum);
    }
    return 0;
}
```

```
;10783 - Odd Sum

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
```

```
I DW 0
J DW 0
SUM1 DW 0
PUT DW 0
COUNT DW 0

.CODE

MAIN PROC
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT
   XOR AX,AX
   XOR BX,BX
   XOR CX ,CX
   XOR DX,DX

   MOV B,2

   CALL SCAN_NUM   ;INPUT TESTCASE
   MOV DX,CX
   PRINTN
    ADD N,DX
   MOV I,1

   FOR1:
     CMP N,1
     JE EXIT

   CALL SCAN_NUM ; input a
   MOV BX,CX
   ADD PUT,BX

   PRINTN

   CALL SCAN_NUM  ;input b
   MOV CX,CX
   printn

  FOR:

    CMP PUT,CX
    JLE CAL
     JMP PRINTF

  CAL:
     XOR DX,DX
     XOR AX,AX
     ADD AX,PUT
    DIV B
    CMP DX,0
    JNE SUM
     INC PUT
     XOR BX,BX
     ADD BX,PUT
```

```
    JMP FOR

  SUM:
    ADD COUNT,BX
    INC PUT
    XOR BX,BX
    ADD BX,PUT
    JMP FOR

 PRINTF:
     XOR AX,AX
     MOV AX,COUNT
     CALL PRINT_NUM
     DEC N
     JMP FOR1
  EXIT:


MAIN ENDP

  DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS
  END MAIN
//uva prblm 11332
#include<stdio.h>
int sum(int n);
int main()
{
  int a,b,c,n;

  while(scanf("%d",&n)==1){
  a = sum(n);
  b = sum(a);
  c = sum(b);
  printf("%d\n",c);}
  return 0;

}
int sum(int n)
{
   int sum=0,digit;
   while(n!=0)
   {
     digit =n%10;//123%10=3
      sum= sum+digit;
      n=n/10;//123/10=12

   }
   return sum;
}
;uva prblm 11332

.MODEL SMALL
.STACK 100H
.DATA
DIGIT DW 0
N1 DW 0
```

```
N DW 0
N3 DW 0
A DW 0
B DW 0
C DW 0
Z DW 3

.CODE

MAIN PROC
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   XOR AX,AX
   XOR BX,BX  ;CLEAR REGISTER
   XOR CX,CX
   XOR DX,DX

    CALL SCAN_NUM

    PRINTN
    MOV AX,CX
    MOV N,AX
    XOR AX,AX

    CALL SUM   ;A=SUM(N)
    MOV AX,C

   MOV N,0
    MOV N,AX
    XOR AX,AX

    CALL SUM
    MOV AX,C  ;B=SUM(A)

     MOV N,0
    MOV N,AX
    XOR AX,AX

    CALL SUM
    MOV AX,C  ;C=SUM(B)

    CALL PRINT_NUM
    PRINTN

MAIN ENDP

    SUM PROC
```

```
    MOV A,10
    MOV C,0

   WHILE:
       CMP N,0
       JE EXIT

       MOV AX,N
      DIV A
      ADD C,DX
      MOV N,AX
      XOR AX,AX
      XOR DX,DX
      JMP WHILE
   EXIT:
     RET

     SUM ENDP
```

```
DEFINE_SCAN_NUM
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNS


END MAIN
```

```
//UVA PRBLM 11854
#include<stdio.h>
#include<math.h>
int main()
{
   int a,b,c,d,i,n;
   while(scanf("%d%d%d",&a,&b,&c)==2)
   {
      if(a==0&&b==0&&c==0)
         break;
      else
      {
         if(c==sqrt((a*a)+(b*b)))
            printf("right\n");
         else
            printf("wrong\n");

      }
   }
   return 0;
}
```

```
;UVA PRBLM 11854

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
B DW 0
C DW 0
```

```
.CODE
MAIN PROC

    STRT:
    MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
    MOV DS,AX

    INCLUDE 'EMU8086.INC'

    ;FOR USER INPUT

    DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS


    XOR CX,CX
    XOR AX,AX
    XOR BX,BX  ;O ASIGN ALL THE RESISTER
    XOR DX,DX

   ; PRINTN "ENTER A"

    CALL SCAN_NUM

    MOV AX,CX
    ADD A,AX
    IMUL A
    MOV A,AX
    CMP A,0  ;A==0 THEN EXIT
    JE EXIT
    XOR AX,AX

    ;PRINTN "ENTER B"
    CALL SCAN_NUM
    MOV AX,CX
    ADD B,AX
    IMUL B
    MOV BX,AX

    CMP BX,0
    JE EXIT
    XOR AX,AX

    ADD BX,A

    ;PRINTN "ENTER C"
    CALL SCAN_NUM
    MOV AX,CX
    ADD C,AX
    IMUL C
    MOV DX,AX

    CMP DX,0
    JE EXIT
```

```
    XOR AX,AX



    CMP BX,DX ;C2==A2 *B2
    JE PRINTR

    PRINTN "WRONG"
    JMP EXIT

    PRINTR:
    PRINTN "RIGHT"




     EXIT:

        MOV AH,4CH
        INT 21H

END MAIN
//uva 10035
#include<stdio.h>
int main()
{
    int a,b,r1,r2;
    while(1)
    {
int sum=0,c=0;
    scanf("%d%d",&a,&b);
    if(a==0&&b==0)
        break;
    while(a>0&&b>0)
    {


        r1=a%10;

        r2=b%10;

        sum = r1+r2;
        a=a/2;
        b=b/2;
        if(sum+c>=10)
        {
            c++;
        }

    }
    if(c>0)
    {
        printf("%d carry operation\n",c);
    }
    else
        printf("No carry operation \n");
  }
  return 0;
}
```

```
;UVA PRBLM 10035

.MODEL SMALL
.STACK 100H
.DATA
SUM DW 0
A DW 0
B DW 0
R1 DW 0
R2 DW 0
C dw ?
D DW 0
E DW 0
P DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

    INCLUDE 'EMU8086.INC'


   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

   MOV D,10
   MOV E,2

   CALL SCAN_NUM
   ;ADD P,CX
   MOV A,CX

   PRINTN

   CALL SCAN_NUM
   ; ADD Q,CX
   MOV B,CX
   PRINTN

   CMP A,0
   JE EXIT

   CMP B,0
   JE EXIT

   WHILE:
   CMP A,0
   JG CHECK

  CHECK:
    CMP B,0
    JG CAL
     JMP IF
```

```
CAL:
  XOR AX,AX
  ADD AX,A
  DIV D
  MOV R1,DX
  XOR DX,DX  ;R1
  XOR AX,AX

  ADD AX,B
  DIV D
  MOV R2,DX ;R2
  XOR DX,DX
  XOR AX,AX

  MOV AX,R1   ;R2+R1
  ADD AX,R2
  MOV SUM,AX

  XOR AX,AX


  MOV AX,A

  DIV E
  MOV A,AX
   XOR AX,AX   ;A/2
  XOR DX,DX


  MOV AX,B
  DIV E
  MOV B,AX
  XOR AX,AX
  XOR DX,DX  ;B/2

  MOV AX,SUM
  ADD AX,C

  CMP AX,10
  JGE L2
  JMP WHILE

  L2:

   INC C

   JMP WHILE


  IF:
    CMP C,0
    JG PRINT1

  PRINT1:
     MOV AX,C
     CALL PRINT_NUM

     PRINTN "CARRRY OPERATION"
```

```
        JMP EXIT

     PRINT2:
         PRINTN " NO CARRY OPERATION"


   EXIT:

      MOV AH,4CH
      INT 21H


  MAIN ENDP



 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN

//uva 12289
#include<stdio.h>
#include<string.h>
int main()
{
   int n;
   scanf("%d",&n);
while(n--)
{
    char a[10];
scanf("%s",a);

 if(strlen(a)==5)
   printf("3\n");
 else
 {
int c=0;
    if((a[0])=='o')
      c++;
    if((a[1])=='n')
      c++;
    if((a[2])=='e')
      c++;
    if(c>=2)
      printf("1\n");
    else
      printf("2\n");}}
    return 0;
}

;#UVA PRBLM 1124

.MODEL SMALL
```

```
.STACK 100H
.DATA
char db 30 DUP(?)
;ARRAY1 DB 100 DUB(?)
A DB 0
COUNT DB 0
COUNT1 DB 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT



        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX

        MOV BX,0
         MOV COUNT,0

       Input:       ;get the string
         mov ah, 1  ;input
         int 21h
        cmp al, 13

        je after    ;if enter then exit

        mov char[bx], al
           inc bx

           INC COUNT
            PRINTN
        jmp Input   ;jumpt to input

    after:
      CMP COUNT,5
      JE PRINT3

        XOR BX,BX
        MOV BX,0
      MOV AL,CHAR[BX]
      CMP AL,'O'
      JE L2

        INC BX

        MOV AL,0
        MOV AL,CHAR[BX]
        CMP AL,'N'
        JE L2
```

```
    INC BX
    MOV AL,0
    MOV AL,CHAR[BX]
    CMP AL,'E'
    JE L2
    JMP IF
L2:
  INC COUNT1

IF:
 CMP COUNT1,2
 JGE PRINT1
 JNGE PRINT2
  JMP EXIT

PRINT1:
  PRINTN "1"
  JMP EXIT

PRINT2:
PRINTN "2"
JMP EXIT

PRINT3:
PRINTN "3"
JMP EXIT
```

```
        EXIT:
        MOV AH,4CH
        INT 21H

MAIN ENDP

DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS
END MAIN
```

```
    //uva 11984
#include<stdio.h>
int main ()
{
  int testCase; scanf ("%d", &testCase);
  int cases = 0;

  while ( testCase-- ) {
    int c, d; scanf ("%d %d", &c, &d);

    double f = (9.0/5.0) * c + 32;
    f += d;
    f -= 32;
    f *= 5;
    f /= 9;

    printf ("Case %d: %.2lf\n", ++cases, f);
  }
```

```
   return 0;
}
;UVA PRBLM 11984

.MODEL SMALL
.STACK 100H
.DATA
TESTCASE DW 0
CASE DW 0
C DW 0
D DW 0
F DW 0
M DW 0
N DW 0
P DW 0
Q DW 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT
```

```
   XOR CX,CX
   XOR BX,BX     ;CLEAR REGISTER
   XOR AX,AX
    XOR DX,DX
    MOV M,9
    MOV N,5
    MOV F,32


CALL SCAN_NUM
MOV TESTCASE,CX
PRINTN

WHILE:
   CMP TESTCASE,0
   JE STRT

   CALL SCAN_NUM
   MOV C,CX
   PRINTN

   CALL SCAN_NUM
   MOV D,CX
   PRINTN

  MOV AX,M
  DIV N
```

```
MOV P,AX
XOR AX,AX
XOR DX,DX

MOV AX,P
MUL C
MOV BX,AX
ADD BX,F
ADD BX,D
SUB BX,F
MOV AX,BX
MUL N
XOR DX,DX
DIV M

PRINT "CASE: "
CALL PRINT_NUM
PRINTN

XOR DX,DX
XOR AX,AX



    JMP STRT
```

```
        EXIT:
        MOV AH,4CH
        INT 21H

MAIN ENDP

DEFINE_SCAN_NUM
  DEFINE_PRINT_NUM
  DEFINE_PRINT_NUM_UNS
END MAIN
```

```
    #include<stdio.h>
int main()
{
  int a;
  int n;
  while(1)
  {
   scanf("%d",&a);
  if(a==0)
    break;
  else
  {
    if(a%11==0)
      printf("%d is a multiple of 11.\n",a);
    else
      printf("%d is not a multiple of 11.\n",a);
  }
```

```
   }
   return 0;
}
;UVA PRBLM 10970

.MODEL SMALL
.STACK 100H
.DATA
M DW 0
N DW 0
A DW 0
.CODE
MAIN PROC

   STRT:
   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
```

```
XOR CX,CX
XOR AX,AX
XOR BX,BX  ;O ASIGN ALL THE RESISTER
XOR DX,DX

MOV N,11

CALL SCAN_NUM
ADD A,CX
MOV M,CX

PRINTN
CMP M,0
JE EXIT
JMP CALCULATION

CALCULATION:
      MOV AX,M
       DIV N
       CMP DX,0
       JE PRINT1
       JNE PRINT2

PRINT1:
  XOR DX,DX
  XOR AX,AX
  MOV AX,A
 CALL PRINT_NUM
 PRINTN "IS A MULTIPLE OF 11."
```

```
    JMP STRT

  PRINT2:
  XOR DX,DX
  XOR AX,AX
    MOV AX,A
    CALL PRINT_NUM
    PRINTN " :IS NOT A MULTIPLE OF 11."



          JMP STRT



   EXIT:

      MOV AH,4CH
      INT 21H

;
END MAIN
```

```
//UVA PRBLM 10079

#include<stdio.h>
int main()
{
   int p;
   long long int n;
   while(scanf("%lld",&n))
   {
      if(n<0)
         break;
      printf("%lld\n",1+ (n*(n+1)/2));
   }
   return 0;
}
;UVA PRBLM 10079


.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
```

```
MOV DS,AX     ;DATA INITIALIZATION

INCLUDE 'EMU8086.INC'
;FOR USER INPUT

    XOR CX,CX
    XOR BX,BX      ;CLEAR REGISTER
    XOR AX,AX
     XOR DX,DX
     MOV BX,2

     CALL SCAN_NUM  ;SCAN N
     PRINT

     MOV AX,CX
     ADD A,AX

     CMP AX,0
     JE EXIT  ;N<0 BREAK

     ADD AX,1 ;ELSE N+1

     MUL A     ;N*N+1

     DIV BX         ;/2

                    ADD AX,1   ;+1

                    PRINTN " "

                    CALL PRINT_NUM


                      JMP STRT

                    EXIT:
                    MOV AH,4CH
                    INT 21H

                  MAIN ENDP

                 DEFINE_SCAN_NUM
                   DEFINE_PRINT_NUM
                   DEFINE_PRINT_NUM_UNS
                END MAIN
                     //UVA PRBLM 11498
                #include<stdio.h>
                int main()
                {
                   int n,m,i,x,y,k;
                   while(scanf("%d",&k)==1)
                   {
```

```
    if(k==0)
       break;
    scanf("%d%d",&n,&m);
    for(i=0; i<k; i++)
    {
       scanf("%d%d",&x,&y);
       if(n==x || m==y)
          printf("divisa\n");
       else if(x>n && y>m)
          printf("NE\n");
       else if(x<n&&y>m)
       {
          printf("NO\n");
       }
       else if(x<n&&y<m)
       {
          printf("SO\n");
       }
       else if(x>n&&y<m)
       {
          printf("SE\n");
       }

    }
  }
  return 0;
}
;#UVA PRBLM 11498
```

```
.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
M DW 0
I DW 0
K DW 0
X DW 0
Y DW 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION



   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT



        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX
```

```
CALL SCAN_NUM
MOV K,CX
PRINTN        ;CX = K

CMP K,0
JE EXIT

CALL SCAN_NUM
MOV N,CX
PRINTN

CALL SCAN_NUM
MOV M,CX
PRINTN

MOV DX,0  ;I=0=DX

LOOP1:
   CMP DX,K
   JG EXIT

   CALL SCAN_NUM
   MOV AX,CX      ;X=AX
   PRINTN

   CALL SCAN_NUM
   MOV BX,CX      ;Y=BX

CMP AX,N
JE CHECK1   ;X== N

CMP AX,N
JG CHECK2   ;X>N

CMP AX,N   ;X<N
JL CHECK3

CMP AX,N
JL CHECK4  ;X<N

CHECK1:
    CMP BX,M  ;Y==M
    JE PRINT1

CHECK2:
     CMP BX,M
     JG PRINT2  ;Y>M

CHECK3:
      CMP BX,M
      JG PRINT3
CHECK4:
    CMP BX,M
    JL PRINT4

PRINT1:
    PRINTN "DIVISA"
    INC DX
```

```
    JMP LOOP1

PRINT2:
    PRINTN "NE"
    INC DX
    JMP LOOP1

PRINT3:
    PRINTN "NO"
    INC DX
    JMP LOOP1

PRINT4:
    PRINTN "SO"
    INC DX

    JMP LOOP1

    EXIT:
    MOV AH,4CH
    INT 21H

MAIN ENDP
 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
      //UVA PRBLM 12646
#include <stdio.h>
int main()
{
int a,b,c;
while(scanf("%d %d %d", &a, &b, &c) == 3)
{
   if(a == b && b == c)
    printf("*\n");
   else if(a != b && b == c)
        printf("A\n");
   else if(a != b && a == c)
        printf("B\n");
   else if(a == b && a != c)
        printf("C\n");
}
return 0;
}
;UVA PRBLM 12646

.MODEL SMALL
.STACK 100H
.DATA
V DW 0
```

```
T DW 0
.CODE
MAIN PROC


   MOV AX,@DATA  ;DATA SEGMENT INITIALIZE
   MOV DS,AX

   INCLUDE 'EMU8086.INC'

   ;FOR USER INPUT


   XOR CX,CX
   XOR AX,AX
   XOR BX,BX  ;O ASIGN ALL THE RESISTER
   XOR DX,DX

  CALL SCAN_NUM
            ;SCAN A
  MOV AX,CX
   PRINTN
 ; XOR CX,CX

  CALL SCAN_NUM
             ;SCAN B
  MOV BX,CX
   PRINTN
  ;XOR CX,CX
   CALL SCAN_NUM     ;SCAN C
   PRINTN
  MOV  DX,CX


LEVEL1:
     CMP AX,BX
     JE LEVEL3    ;AX==BX
     JNE LEVEL8

LEVEL2:
     CMP AX,DX    ;AX==DX
     JE LEVEL4


LEVEL3:
      CMP BX,DX  ;BX==DX
      JE PRINT1

LEVEL4:
      CMP AX,BX
      JNE PRINT3 ;AX!=BX

LEVEL5:
     CMP AX,DX     ;AX!=DX
     JNE LEVEL7

LEVEL6:
```

```
    CMP BX,DX
    JE PRINT2  ;BX==DX
LEVEL7:
    CMP AX,BX
    JE PRINT4   ;AX==BX
LEVEL8:
    CMP AX,DX    ;AX==DX
     JE PRINT3


 PRINT1:
       PRINTN "*"
       JMP EXIT


 PRINT2:
      PRINTN "A"
      JMP EXIT


 PRINT3:
      PRINTN "B"
      JMP EXIT


  PRINT4:
       PRINTN "C"
       JMP EXIT




  EXIT:
       MOV AH,4CH
       INT 21H



    MAIN ENDP

   DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS


;
END MAIN
//uva 12917
    #include<stdio.h>
int main()
{
   int x,y,z;
   while(scanf("%d%d%d",&x,&y,&z)==3)
   {

      if(x<=(z-y))
         printf("Props win!\n");
      else
         printf("Hunters win!\n");
   }
   return 0;
}
;#UVA PRBLM 12917
```

```
.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT



        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX

        CALL SCAN_NUM
        MOV DX,CX      ;X INPUT
        PRINTN

        CALL SCAN_NUM
        MOV BX,CX      ;Y INPUT
        PRINTN

        CALL SCAN_NUM
        MOV AX,CX       ;Z INPUT
        PRINTN

        SUB AX,BX
        MOV A,AX    ;Z-Y

        CMP DX,A
        JLE PRINT12

        PRINTN "Hunters win!"
        jmp strt

        PRINT12:
           printn "Props win!"

           jmp strt


           EXIT:
           MOV AH,4CH
           INT 21H
```

```
  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
//uva 12952
      #include<stdio.h>
int main()
{
   int a,b;
   while( scanf("%d%d",&a,&b)!=EOF)
   {
      if(a>=b)
         printf("%d\n",a);
      else
         printf("%d\n",b);
   }
   return 0;

}
;#UVA PRBLM 12952

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW ?
R DW ?
K DW ?
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT


        XOR CX,CX
        XOR BX,BX   ;CLEAR REGISTER
        XOR AX,AX
         XOR DX,DX


         CALL SCAN_NUM  ;DX = A
         MOV DX,CX
         PRINTN

         CALL SCAN_NUM
         MOV BX,CX     ;BX = B
         PRINTN
```

```
        CMP DX,BX
        JGE L1

        L2:
          MOV AX,BX
          CALL PRINT_NUM
          JMP EXIT
        L1:
          MOV AX,DX
          CALL PRINT_NUM



           EXIT:
           MOV AH,4CH
           INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
//UVA PRBLM 12992
#include<stdio.h>
int main()
{
   int n,m,count = 1;
   scanf("%d",&n);
   while(n--)
   {

      scanf("%d",&m);
       printf("Case #%d: %d\n",count++,2*m-1);
   }
   return 0;
}
;#UVA PRBLM  12992

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
I DW 0
J DW 0
.CODE
MAIN PROC
   STRT:
   MOV AX,@DATA
   MOV DS,AX     ;DATA INITIALIZATION


   INCLUDE 'EMU8086.INC'
   ;FOR USER INPUT
```

```
    XOR CX,CX
    XOR BX,BX   ;CLEAR REGISTER
    XOR AX,AX
     XOR DX,DX
     MOV B,2
CALL SCAN_NUM
MOV BX,CX
mov a,0
PRINTN

WHILE:
    CMP BX,A
    JE EXIT

    CALL SCAN_NUM
    MOV AX,CX
    PRINTN

    MUL B
    SUB AX,1
    MOV DX,AX
     INC A

    PRINT "Case #"
     MOV AX,A
     CALL PRINT_NUM
     PRINT ":"
                    MOV AX,DX
                    CALL PRINT_NUM
                    PRINTN

                    JMP WHILE

                    EXIT:
                    MOV AH,4CH
                    INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
    DEFINE_PRINT_NUM
    DEFINE_PRINT_NUM_UNS
END MAIN
       //UVA PRBLM 13012
#include<stdio.h>
int main()
{
    int n,m,i,count;
    while(scanf("%d",&n)!=EOF)
```

```
  {
     count =0;


        for(i=0;i<5;i++){
        scanf("%d",&m);
        if(n==m)
        count++;

        }


     printf("%d\n",count);
  }
  return 0;
}
```

```
;#UVA PRBLM  13012

.MODEL SMALL
.STACK 100H
.DATA
A DW 0
N DW 0
B DW 0
I DW 0
J DW 0
.CODE
MAIN PROC
   STRT:
MOV AX,@DATA
MOV DS,AX     ;DATA INITIALIZATION


INCLUDE 'EMU8086.INC'
;FOR USER INPUT


      XOR CX,CX
      XOR BX,BX   ;CLEAR REGISTER
      XOR AX,AX
       XOR DX,DX

    call scan_num
    MOV BX,CX
    PRINTN
    MOV A,0


    CAL:
      CMP BX,A
      JE PRINT1
       INC A
      CALL SCAN_NUM
      MOV DX,CX
      PRINTN
```

```
        CMP DX,BX
        JE LOOP1
        JMP CAL


      LOOP1:
        INC J
        JMP CAL

        PRINT1:
            MOV AX,J
            ;PRINTN
            CALL PRINT_NUM
            PRINTN


            EXIT:
            MOV AH,4CH
            INT 21H

  MAIN ENDP

 DEFINE_SCAN_NUM
   DEFINE_PRINT_NUM
   DEFINE_PRINT_NUM_UNS
END MAIN
```