31: UVA_458_Assembly

```
include 'emu8086.inc'
.model small
.stack 100h
.data
   char db 1000 DUP(?)
.code
main proc
  mov ax, @data
  mov ds, ax
  mov bx, 0  ;bx=0
Input:       ;get the string
  mov ah, 1  ;input
  int 21h
  cmp al, 13
  je after    ;if enter then exit
  sub al, 7
  mov char[bx], al
  inc bx      ;bx++
  jmp Input   ;jumpt to input

after:
   mov si, 0  ;si=0
   printn     ;newline
 Output:      ;output
   cmp si, bx ;si=bx then exit
   je exit
   mov dl, char[si]
   mov ah, 2
   int 21h
   inc si      ;si++
   jmp output

exit:
   mov ah, 4ch
   int 21h
   main endp
end main
```

31: UVA_458_C

```
#include<stdio.h>

int main()
{
   char ch[10000];
   int length, i;
   while(gets(ch))
   {
      length=strlen(ch);
      for(i=0; i<length; i++)
      {
         ch[i]=ch[i]-7;
      }
      puts(ch);
   }
   return 0;
}
```

32. UVA_494_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data                   ; data section
   char db 100 dup(?)
   length dw ?
   word dw ?
   count dw ?
   i dw ?
   char1 db ?
.code
main proc

   mov ax, @data
   mov ds, ax

 while:
   mov bx, 0
  Input_string:     ; input string
      mov ah, 1
      int 21h
      mov char1, al
      cmp al, 13      ;if press enter then exit from
input
      je exit_input_string
      mov al, char1
      mov char[bx], al ; insert into array
      inc bx
  jmp Input_string
exit_input_string:
      printn
      mov length, bx ; length=bx
      mov count, 0   ;count=0
      mov word, 1    ;word=1
      mov i, 0       ;i=0
      mov bx, 0      ;bx=0
for:
    mov ax, i
    cmp ax, length
    je exit_for       ;if i = length hen exit
                ;if((ch[i]>='A' && ch[i]<='Z')
    mov al, char[bx]
    cmp al, 65
    jl check_az
    cmp al, 90
    jg check_az

 if:            ;if(word)
   mov cx, word
   cmp cx, 1
   jne after
   inc count     ;count++
after:
   mov word, 0   ;word=0
   jmp for_last  ;jump to last

check_az:          ;if(ch[i]>='a' && ch[i]<='z')
   mov al, char[bx]
   cmp al, 97
   jl else
   cmp al, 122
   jg else
   jmp if      ; jump to if

else:         ;word=0
   mov word, 1

for_last:
   inc bx    ;bx++
   inc i     ;i++
 jmp for
exit_for:
 mov ax, count  ;ax=count
 call print_num  ;output count
 printn
 jmp while        ;jump to while

exit:
      mov ah, 4ch
      int 21h
   main endp
define_print_num    ; define build in function
define_scan_num
define_print_num_uns
end main
```

32. UVA_494_C

```
#include<stdio.h>
int main()
{
   char ch[10000];
   int i, count, word;
   while(gets(ch))
   {
      count=0;
      word=1;

      for(i=0; ch[i]!='\0'; i++)
      {
         if((ch[i]>='A' && ch[i]<='Z') || (ch[i]>='a'
&& ch[i]<='z'))
         {
            if(word){
            count++;
            }
            word=0;
         }

         else
         {
            word=1;
         }
      }
      printf("%d\n", count);
   }
   return 0;
}
```

33. UVA_913_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data
   n dw ?
   sum dw ?
   odd dw ?
.code
main proc
   mov ax, @data
   mov ds, ax
while:
   call scan_num    ; input the number
   printn
   mov ax, cx       ;ax=cx
   mov n, cx        ;n=cx
   add ax, 2        ;n+2
   mov cx, n
   mul cx          ;n*(n+2)
   mov cx, 2       ;cx=2
   div cx          ;ax/cx
   mov odd, ax     ;odd=ax
   mov cx, 3       ;cx=3
   mul cx          ;ax=ax*cx
   sub ax, 6       ;ax-ax
   call print_num  ;print the number
   printn
jmp while
   main endp
                ; define buid in function
define_scan_num
define_print_num
define_print_num_uns
end main
```

33. UVA_913_C

```
#include <stdio.h>
int main()
{
   long int n,sum,odd;
   while(scanf("%ld",&n)==1)
   {
   odd=(n*(n+2))/2;
   sum=(3*odd)-6;
   printf("%ld\n",sum);
   }
return 0;
}
```

34.UVA_1124_ASM

```
include 'emu8086.inc'
.model small
.data
    str db 100 DUP (?)
    char db ?
.code
main proc

while:
   mov bx, 0
for:
    mov ah, 1   ;input from user
    int 21h
    mov char, al ;char=al
    cmp al, 13
    je exit_for  ;if enter then exit
    mov al, char
    mov str[bx], al ;str[bx]=char
    inc bx        ;bx++
    jmp for       ;jump to for
exit_for:
   printn         ;newline
   mov si, bx     ; si=bx
   mov bx, 0      ;bx=0
for2:
    mov cx, bx
    cmp cx, si    ;compare cx, si
    je after      ; jump equal
    mov ah, 2
    mov dl, str[bx] ; output
    int 21h
    inc bx
    jmp for2       ; jump to for2

after:
    printn
    jmp while      ; jump to while
Exit:
    mov ah, 4ch
    int 21h
    main endp
end main
```

34.UVA_1124_C

```
#include <stdio.h>
int main()
{
    char a[100];
    while(gets(a))
    puts(a);
    return 0;
}
```

35.UVA_10082_ASM

```
include 'emu8086.inc'  ; include library function
.model small
.stack 100h
.data            ; data section
   char db 1000 dup(?)
   length dw ?
   i dw ?

.code            ; code section
main proc        ; main proc start
   mov ax, @data ; import data
   mov ds, ax

while:

   mov bx, 0
string_input:    ; input string
   mov ah, 1
   int 21h
   cmp al, 13
   je exit_s_i
   char[bx], al
   inc bx
   jmp string_input
exit_s_i:
   mov length, bx    ;length=bx
   mov i, 0          ;i=0
 for:
   mov ax, i         ;ax=i
   cmp ax, length    ;compare ax, length
   je exit_for       ;switch, case

   mov al, s[bx]
   cmp al, 'W'
   je pQ

   mov al, s[bx]
   cmp al, 'E'
   je pW

    mov al, s[bx]
   cmp al, 'R'
   je pE

    mov al, s[bx]
   cmp al, 'T'
   je pR

    mov al, s[bx]
   cmp al, 'Y'
   je pT

    mov al, s[bx]
   cmp al, 'U'
   je pY

    mov al, s[bx]
   cmp al, 'I'
   je pU

    mov al, s[bx]
   cmp al, 'O'
   je pI

    mov al, s[bx]
   cmp al, 'P'
   je pO


    mov al, s[bx]
   cmp al, 'S'
   je pA

    mov al, s[bx]
   cmp al, 'D'
   je pS

    mov al, s[bx]
   cmp al, 'F'
   je pD

    mov al, s[bx]
   cmp al, 'G'
   je pF

    mov al, s[bx]
   cmp al, 'H'
   je pG

    mov al, s[bx]
   cmp al, 'J'
   je pH

    mov al, s[bx]
   cmp al, 'K'
   je pJ


    mov al, s[bx]
   cmp al, 'L'
```

```
je pK

 mov al, s[bx]
cmp al, 'X'
je pZ

 mov al, s[bx]
cmp al, 'C'
je pX

 mov al, s[bx]
cmp al, 'V'
je pC

 mov al, s[bx]
cmp al, 'B'
je pV


 mov al, s[bx]
cmp al, 'N'
je pB

 mov al, s[bx]
cmp al, 'M'
je pN

 mov al, s[bx]
cmp al, '2'
je p1

 mov al, s[bx]
cmp al, '3'
je p2


 mov al, s[bx]
cmp al, 4
je p3

 mov al, s[bx]
cmp al, 5
je p4


 mov al, s[bx]
cmp al, 6
je p5

 mov al, s[bx]
cmp al, 7
je p6


 mov al, s[bx]
cmp al, 8
je p7

 mov al, s[bx]
cmp al, 9
je p8

 mov al, s[bx]
cmp al, 0
je p9

 mov al, s[bx]
cmp al, '-'
je p0

 mov al, s[bx]
cmp al, '='
je pHypen


 mov al, s[bx]
cmp al, '['
je pP


 mov al, s[bx]
cmp al, ']'
je pSthird

 mov al, s[bx]
cmp al, '\'
je pFthird

 mov al, s[bx]
cmp al, ';'
je pL

 mov al, s[bx]
cmp al, 1
je pBef1

 mov al, s[bx]
cmp al, ','
je pM

 mov al, s[bx]
```

```
   cmp al, '.'
   je pComma

    mov al, s[bx]
   cmp al, '/'
   je pDot

    mov al, s[bx]
   cmp al, ' '
   je pSpace

   print ";"
   jmp after

 pS:
   print "S"
   jmp after
 pD:
   print "D"
   jmp after
pF:
   print "F"
   jmp after
 pG:
   print "G"
   jmp after

 pA:
   print "A"
   jmp after
 pO:
   print "O"
   jmp after
  pI:
   print "I"
   jmp after
  pU:
   print "U"
   jmp after
 pY:
   print "Y"
   jmp after
 pT:
   print "T"
   jmp after
 pR:
   print "R"
   jmp after
 pE:
   print "E"
   jmp after
```

```
 pK:
   print "K"
   jmp after

 pH:
   print "H"
   jmp after

 pZ:
   print "Z"
   jmp after

pJ:
   print "J"
   jmp after

 pV:
   print "V"
   jmp after


 pX:
   print "X"
   jmp after

pC:
   print "C"
   jmp after

pB:
   print "B"
   jmp after

pN:
   print "N"
   jmp after

p1:
   print "1"
   jmp after

p2:
   print "2"
   jmp after


p3:
   print "3"
   jmp after
```

```
p4:
   print "4"
   jmp after


p5:
   print "5"
   jmp after


p6:
   print "6"
   jmp after

p7:
   print "7"
   jmp after



p8:
   print "8"
   jmp after

p9:
   print "9"
   jmp after

p0:
   print "0"
   jmp after


pHypen:
   print "-"
   jmp after

pP:
   print "P"
   jmp after
pSthird:
   print "["
   jmp after

pFthird:
   print "]"
   jmp after
pW:
   print "W"
   jmp after
```

```
pL:
   print "L"
   jmp after

pBef1:
   print "`"
   jmp after

pM:
   print "M"
   jmp after

pComma:
   print ","
   jmp after

pdot:
   print "."
   jmp after
pSpace:
   print " "
   jmp after

 pQ:
   print "Q"
   jmp after


 after:
   printn
   int i
 jmp for

exit_for:
   jmp while

exit:
   mov ah, 4ch   ; exit
   int 21h
   main endp      ; main proc end
   define_print_num_uns
   define_print_num
   define_scan_num
end main
```

35.UVA_10082_C

```
#include<stdio.h>
#include<string.h>
int main(){
long int length,i; char s[100000];
while(gets(s)){
   length=strlen(s);
   for(i=0;i<length;i++){
      switch(s[i]){
         case 'W':
          printf("Q");    break;
         case 'E':
          printf("W"); break;
         case 'R':
          printf("E");    break;
         case 'T':
          printf("R");    break;
         case 'Y':
          printf("T");    break;
         case 'U':
          printf("Y");    break;
         case 'I':
          printf("U");    break;
         case 'O':
          printf("I");    break;
         case 'P':
          printf("O");    break;
         case 'S':
          printf("A");    break;
         case 'D':
          printf("S");    break;
         case 'F':
          printf("D");    break;
         case 'G':
          printf("F");    break;
         case 'H':
          printf("G");    break;
         case 'J':
          printf("H");    break;
         case 'K':
          printf("J");   break;
         case 'L':
          printf("K");    break;
         case 'X':
          printf("Z");    break;
         case 'C':
          printf("X");    break;
         case 'V':
          printf("C");    break;
         case 'B':
          printf("V");    break;
         case 'N':
          printf("B");    break;
         case 'M':
          printf("N");    break;
         case '2':
          printf("1");    break;
         case '3':
          printf("2");    break;
         case '4':
          printf("3");    break;
         case '5':
          printf("4");    break;
         case '6':
          printf("5");    break;
         case '7':
          printf("6");    break;
         case '8':
          printf("7");    break;
         case '9':
           printf("8");    break;
         case '0':
          printf("9");    break;
         case '-':
          printf("0");    break;
         case '=':
          printf("-");    break;
         case '[':
          printf("P");    break;
         case ']':
          printf("[");    break;
         case '\\':
          printf("]");    break;
         case ';':
          printf("L");    break;
         case '1':
          printf("`");    break;
         case ',':
          printf("M");    break;
         case '.':
          printf(",");    break;
         case '/':
          printf(".");    break;
         case ' ':
          printf(" ");    break;
         default:
          printf(";");    break;
          }       }
    printf("\n"); }
return 0;
}
```

36.UVA_10469_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data                   ; data section
   a dw ?
.code
main proc
 while:
   call scan_num
   mov a, cx          ;cin>>a
   printn
   call scan_num   :cin>>b
   printn

   xor cx, a          ;a xor b
   mov ax, cx
   call print_num
   printn
   jmp while

   mov ah,4ch
   int 21h

   main endp
                ; define build in function
define_scan_num
define_print_num
define_print_num_uns
end main
```

36.UVA_10469_C

```
#include <bits/stdc++.h>
using namespace std;


int main()
{
   int a,b;

   while(cin >> a >> b)
   {
      cout << (int)(a^b) << endl;
   }
   return 0;
}
```

37.UVA_10696_Asm

```
include 'emu8086.inc'  ; include library function
.model small
.stack 100h
.data            ; data section
 N dw ?

.code            ; code section
main proc        ; main proc start
   mov ax, @data ; import data
   mov ds, ax

  while:
       call scan_num  ;input N
       mov N, cx
       printn
       cmp cx, 0   ;compare cx and 0
       je exit

       cmp cx, 100
       jg if       ; if greater than 100 then jump to if
       print "f91("
       mov ax, N
       call print_num
       printn ") = 91"  ;printf("f91(%d) = 91\n", N);
       jmp while

  if:              ;printf("f91(%d) = %d\n", N, N-10);
       print "f91("
       mov ax, N
       call print_num
       print ") = "
       mov ax, N
       sub ax, 10
       call print_num
       printn

   jmp while

exit:
   mov ah, 4ch   ; exit
   int 21h
   main endp     ; main proc end
   define_print_num_uns
   define_print_num
   define_scan_num
end main
```

37.UVA_10696_C

```
#include<stdio.h>

int main()
{
   int N;
   while(scanf("%d", &N)==1)
   {
      if(N==0)
         break;

     if(N>100)
        printf("f91(%d) = %d\n", N, N-10);
 else
           printf("f91(%d) = 91\n", N);
      }
   }
   return 0;
}
```

38.UVA_11498_Asm

```
include 'emu8086.inc'
.model small
.data
   n dw ?
   px dw ?
   py dw ?
   x dw ?
   y dw ?
.code
main proc
   call scan_num  ;cin>>n
   mov n, cx
   printn         ;newline
 while:
   mov cx, n     ; compare n and 0
   cmp cx, 0
   je Exit

   call scan_num
    mov px, cx    ; input px
    printn

    call scan_num ; input py
    mov py, cx
    printn

 while2:
    mov cx, n
    cmp cx, 0
    je after2

    call scan_num
    mov x, cx       ; input x
    printn

    call scan_num
    mov y, cx      ;input y
    printn
                ;if (x == px || y == py)
    mov ax, x
    cmp ax, px
    je print_divisa
    mov ax, y
    cmp ax, py
    je print_divisa
    jmp else_if1

 print_divisa:
    printn "divisa"
    jmp after
 else_if1:          ;else if (x < px && y > py)
    mov ax, x
    cmp ax, px
    jl check_py
    jmp elseif2

check_py:
    mov ax, y
    cmp ax, py
    jg print_NO
    jmp elseif2

print_NO:
    printn "NO"
    jmp after

elseif2:        ;else if (x > px && y > py)
   mov ax, x
   cmp ax, px
   jg check_py2
   jmp elseif3

check_py2:
   mov ax, y
   cmp ax, py
   jg print_NE
   jmp elseif3
print_NE:
   printn "NE"
   jmp after

elseif3:        ;else if (x > px && y < py)
   mov ax, x
   cmp ax, px
   jg check_py3
   jmp elseif4

check_py3:
   mov ax, y
   cmp ax, py
   jl print_SE
   jmp elseif4
print_SE:
   printn "SE"
   jmp after

elseif4:          ;else if (x < px && y < py)
   mov ax, x
   cmp ax, px
   jl check_py4
```

```
    jmp after

check_py4:
    mov ax, y
    cmp ax, py
    jl print_SO
    jmp after
print_SO:
    printn "SE"

 after:
     dec n
     printn       ; newline
     jmp while2   ;jump to while

 after2:
    jmp while        ; jump to while

  Exit:
    mov ah, 4ch
    int 21h

    main endp
define_scan_num
define_print_num
define_print_num_uns
 end main
```

38.UVA_11498_C

```
#include <iostream>
using namespace std;

int main() {
	int n, px, py, x, y;

	cin >> n;
	while (n != 0) {
		cin >> px >> py;

		while (n--) {
			cin >> x >> y;

			if (x == px || y == py)
				cout << "divisa";
			else if (x < px && y > py)
				cout << "NO";
			else if (x > px && y > py)
				cout << "NE";
			else if (x > px && y < py)
				cout << "SE";
			else if (x < px && y < py)
				cout << "SO";
			cout << endl;

		}

		cin >> n;
	}
	return 0;
}
```

39.UVA_11526_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data
   T dw ?
   i dw ?
   n dw ?
   res dw ?
   x dw ?
.code
main proc
   mov ax, @data
   mov ds, ax

   call scan_num ; input T
   mov T, cx
   printn       ;newline
 while:
   mov ax, T
   cmp ax, 0   ;compare
   je exit      ; T==0 then exit
   call scan_num ; input x
   mov x, cx
   printn       ;newline

   mov res, 0    ;res=0
   mov i, 1     ;i=1
 for:           ;for(i = 1; i <= n; i++ )
   mov ax, i
   cmp ax, x
   jg exit_for

   xor dx, dx
   mov ax, x
   mov cx, i
   div cx       ;n/i
   add ax, res   ;ax=(res + n/i)
   mov res, ax  ;res = ax
   inc i        ;i++
 jmp for

  exit_for:         ;displying result
       mov ax, res
       call print_num
       printn

   dec T         ; T--
 jmp while         ;jump to while

 Exit:
   mov ah, 4ch
   int 21h

   main endp
            ;define build in function
define_scan_num
define_print_num
define_print_num_uns
end main
```

39.UVA_11526_C

```
#include <stdio.h>

long long H(int n)
{
   long long res = 0;
   int i;
   for(i = 1; i <= n; i++ )
   {
      res = (res + n/i);
   }
   return res;
}

int main()
{
   int T, x, j;
   long long result;
  scanf("%d", &T);
  while(T--)
  {
      scanf("%d", &x);
      result=H(x);
      printf("%lld\n", result);
   }
   return 0;
}
```

40.UVA_11839_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data
   N dw ?
   A dw 6 DUP(?)
   i dw ?
   k dw ?
   j dw ?
   count dw ?
.code
main proc
    mov ax, @data  ; data import
    mov ds, ax

while:
    call scan_num   ;input N
    mov N, cx
    printn
    cmp cx, 0
    jle while

    mov i, 0  ;i=0

 for1:
    mov ax, i
    cmp ax, N     ;compare i and N
    jge exit_for1
    mov count, 0  ;count=0
    mov j, 1
    for2:
      mov ax, j  ;ax=j
      cmp ax, 5  ;ax=5
      jg exit_for2
      call scan_num ; input a number
      printn
      mov bx, j
      mov A[bx], cx  ; insert into A[j]
      cmp cx, 127
      jg after
      inc count     ;count++
      mov ax, j     ; ax=j
      mov k, ax     ;k=ax
   after:
     inc j
    jmp for2

exit_for2:
   mov ax, count  ;ax=count
   cmp ax, 1      ;if(count==1)
   jne else

   mov ax, k
   cmp ax, 1   ;f(k==1)
   je printA
   cmp ax, 2   ;f(k==)
   je printB
   cmp ax, 3   ;f(k==3)
   je printC
   cmp ax, 4   ;f(k==4)
   je printD
   cmp ax, 5    ;f(k==4)
   je printE
   jmp after2
printA:
   printn "A"    ;display A
   jmp after2
printB:
   printn "B"    ;display B
   jmp after2
printC:
   printn "C"    ;display C
   jmp after2
printD:
   printn "D"    ;display D
   jmp after2
printE:
   printn "E"    ;display E
   jmp after2


 else:
     printn "*"  ;display *

after2:
    inc i       ;i++
   jmp for1
 exit_for1:
jmp while      ;jump to while

  Exit:
   mov ah, 4ch
   int 21h
   main endp
                ;define build in function
define_scan_num
define_print_num
define_print_num_uns
end main
```

40.UVA_11839_C

```
#include<stdio.h>
int main()
{
   int N, A[6], i, k, j;
   while(scanf("%d", &N) == 1)
   {
      if(N>0){
      for(i=0; i<N; i++)
      {
         int count=0;
         for(j=1; j<=5; j++)
         {
            scanf("%d", &A[j]);
            if(A[j]<=127)
            {
               count++;
               k=j;
            }
         }
         if(count==1){
         if(k==1)
            printf("A\n");
         else if(k==2)
            printf("B\n");
         else if(k==3)
            printf("C\n");
         else if(k==4)
            printf("D\n");
         else if(k==5)
            printf("E\n");
         }
         else
            printf("*\n");
      }
   }
   }
   return 0;
}
```

41.UVA_11875_Asm

```
include 'emu8086.inc'  ; include library function
.model small
.stack 100h
.data            ; data section
  T dw ?
  i dw ?
  m dw ?
  k dw ?
  A db DUP 100 (0)
  num dw ?

.code            ; code section
main proc        ; main proc start
    mov ax, @data ; import data
    mov ds, ax

while:
    call scan_num   ; input T
    mov T, cx
    printn
    mov m, 1
 for:             ;for start
    mov ax, m       ;ax=m
    cmp ax, T       ;compare m and T
    jg exit_for
    call scan_num   ;input i
    mov i, cx
    printn          ;newline
    mov k, 0        ;k=0
    mov bx, 0       ;bx=0
  for2:
    mov ax, k       ;ax=k
    cmp ax, i       ;compare k and i
    je exit_for2
    call scan_num   ;input A[k]
    printn          ; newline

    mov bx, k       ;bx=k
    mov A[bx], cl   ;insert value into array

    inc k          ;k++
   jmp for2        ;jump to for2
exit_for2:
    printn          ; dispaying result
    print "Case "
    mov ax, m
    call print_num  ; call print_num proc
    print ": "
    mov ax, k
    mov cx, 2
    div cx      ; k/2
    mov bx, ax
    xor ah, ah
    mov al, A[bx]  ; al=A[k/2]
    call print_num
    printn

    inc m       ;m++
 jmp for

 exit_for:

jmp while

exit:
    mov ah, 4ch   ; exit
    int 21h
    main endp      ; main proc end
    define_print_num_uns
    define_print_num
    define_scan_num
end main
```

41.UVA_11875_C

```
#include<stdio.h>
int main(){
 int T,i, m, k;
 int A[100];
   while(scanf("%d", &T)==1)
   {
      for(m=1; m<=T; m++)
      {
         scanf("%d", &i);

         for(k=0; k<i; k++)
         {
            scanf("%d", &A[k]);
         }
         printf("Case %d: %d\n", m, A[k/2]);
      }
   }
   return 0;
}
```

42.UVA_11934_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data           ;data section
   a dw ?
   b dw ?
   c dw ?
   d dw ?
   L dw ?
   i dw ?
   re1 dw ?
   count dw ?
.code               ;code section
main proc
    mov ax, @data
    mov ds, ax
while:
    call scan_num
    mov a, cx       ;input a
    printn
    call scan_num
    mov b, cx       ;input b
    printn
    call scan_num
    mov c, cx       ;input b\c
    printn
    call scan_num
    mov d, cx          ;input d
    printn
    call scan_num
    mov L, cx           ;input L
    printn

    mov ax, a
    cmp ax, 0         ; compare a and 0
    je check2
    jmp else

 check2:
     mov ax, b
    cmp ax, 0
    je check3        ; compare b and 0
    jmp else

  check3:
     mov ax, c
    cmp ax, 0       ; compare c and 0
    je check4
    jmp else

check4:
     mov ax, d
    cmp ax, 0        ; compare d and 0
    je Exit
else:
    mov count, 0    ;count=0
    mov i, 0        ;i=0
  for:
    mov ax, i
    cmp ax, L       ;compare i and L
    jg exit_for
    xor dx, dx      ;dx=0
    mov ax, a
    mov cx, i
    mul cx          ;ax=a*i
    mul cx          ;ax=a*i*i
    mov re1, ax     ; re1=ax

    xor dx, dx      ;dx=0
    mov ax, b
    mov cx, i
    mul cx          ;ax=b*i
    add ax, re1     ;ax=ax+re1
    add ax, c       ;ax+=c
    xor dx, dx      ;dx=0
    mov cx, d       ;cx=d
    div cx          ;ax/cx
    cmp dx, 0       ;compare reminder and 0
    jne after
    inc count
  after:
    inc i       ;i++
    jmp for     ;jump to for
exit_for:

    mov ax, count  ;print value of count
    call print_num
    printn
jmp while           ;jump to while

  Exit:
       mov ah, 4ch
       int 21h
   main endp
            ; define build in function
define_print_num
define_print_num_uns
define_scan_num
end main
```

42.UVA_11934_C

```
#include<stdio.h>
int main()
{
   int a, b, c, d, L, i, f, count;

   while(1)
   {
      scanf("%d%d%d%d%d", &a, &b, &c, &d,
&L);
      if(a==0 && b==0 && c==0 && d==0
&&L==0)
         break;
      else
      {
         count=0;
         for(i=0; i<=L; i++)
         {
            f = a*i*i+b*i+c;
            if(f%d==0)
               count++;
         }
         printf("%d\n", count);
      }
   }
   return 0;
}
```

43.UVA_12289_Asm

```
include 'emu8086.inc'  ; include library function
.model small
.stack 100h
.data            ; data section
   T dw ?
   str_s db 1000 dup (?)
   count dw 0
   length dw 0
.code            ; code section
main proc        ; main proc start
    mov ax, @data ; import data
    mov ds, ax
    call scan_num
    mov T, cx
    printn
while: mov ax, T     ;input t
   cmp ax, 0
   je exit
   mov length, 0 ;length=0
   mov bx, 0
input_string:
   mov ah, 1     ;input string
   int 21h
   cmp al, 13
   je exit_input
   mov str_s[bx], al ; insert data into array
   inc length        ;length++
   inc bx            ;bx++
jmp input_string
   exit_input:
   printn
   mov ax, length
   cmp ax, 5       ;if(s.length() == 5)
   jne else
    printn "3"       ;cout<<3<<endl;
   jmp after
else:  mov bx, 0
    mov al, str_s[bx]
    cmp al, 't'        ;if(s[0]== 't')
    jne if2
    inc count
 if2:  mov bx, 1
    mov al, str_s[bx]
    cmp al, 'w'         ;if(s[1]== 'w')
    jne if3
    inc count
  if3: mov bx, 2
    mov al, str_s[bx]
    cmp al, 'o'       ;if(s[2]=='o')
    jne if4
    inc count
 if4:   mov ax, count
  cmp ax, 2       ; if(count>=2)
  jl else4
  printn "2"       ;print 2
   jmp after
else4:  printn "1"     ;cout<<1<<endl
after: jmp while
exit:
    mov ah, 4ch   ; exit
    int 21h
    main endp      ; main proc end
    define_print_num_uns
    define_print_num
    define_scan_num
end main
```

43.UVA_12289_C

```
#include<bits/stdc++.h>

using namespace std;
int main()
{
    int T;
    string s;
    cin>>T;
    while(T--)
    {
        cin>>s;
        int count=0;
        if(s.length() == 5)
            cout<<3<<endl;
        else
        {
            if(s[0]== 't')
                count++;
            if(s[1]=='w')
                count++;
            if(s[2]=='o')
                count++;
            if(count>=2)
                cout<<2<<endl;
            else
                cout<<1<<endl;
        }
    }
    return 0;
}
```

44.UVA_12403_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data
    t dw ?
    ans dw ?
    char db 100 dup(?)
    num dw ?
.code
main proc
    mov ax, @data
    mov ds, ax

   call scan_num
   mov t, cx         ; input test case
   printn
   mov ans, 0     ;ans=0

 while:
    mov ax, t
    cmp ax, 0    ;if t=0 then exit
    je exit
    mov bx, 0    ;bx=0
 input_string:   ;sting input
    mov ah, 1
    int 21h      ;input letter
    cmp al, 13
    je break_input ; if newline then stop taking input
    cmp al, ' '
    je break_input  ;if space then stop taking input
    mov char[bx], al ;add in array
    inc bx          ;bx++
 jmp input_string

 break_input:
    mov bx, 0     ;bx=0
    cmp char[bx], 'd'  ;if char[0]=d
    je if            ;then jump to if
    mov ax, ans       ; else
    printn           ;newline
    call print_num    ;print ans
    printn           ;newline
    jmp after
if:                 ;if
  print " "
  call scan_num      ;taking new num
  add cx, ans        ; add ans and num
  mov ans, cx        ;ans+=num
  printn            ;newline

after:
    dec t           ;t--
 jmp while


exit:
    mov ah, 4ch
    int 21h
    main endp
            ;define build in function
define_scan_num
define_print_num
define_print_num_uns
end main
```

44.UVA_12403_C

```
#include<bits/stdc++.h>

using namespace std;

int main()
{
    int t;
    cin >> t;
    int ans = 0;
    while(t--)
    {
        string s;
        cin >> s;
        int num;
        if(s[0] == 'd')
        {
            cin >> num;
            ans += num;
        }
        else
            cout << ans << endl;
    }
    return 0;
}
```

45.UVA_12461_Asm

```
INCLUDE 'emu8086.inc'  ; include library
function
.MODEL SMALL
.STACK 100H
.DATA            ; data section
  num dw ?

.CODE            ; code section
MAIN PROC        ; main proc start
   MOV AX, @DATA ; import data
   MOV DS, AX

while:
   call scan_num    ; input n
   printn           ; print newline
   mov num, cx
   cmp cx, 0        ; cmp n and 0
   je Exit          ;if n=0 ten exit
   printn "1/2"     ; print 1/2

 jmp while          ; jmp while

Exit:
   MOV AH, 4CH   ; exit
   INT 21H
   MAIN ENDP      ; main proc end
   DEFINE_PRINT_NUM_UNS   ;define
   DEFINE_PRINT_NUM
   DEFINE_SCAN_NUM
END MAIN
```

45.UVA_12461_C

```
#include<stdio.h>
int main(){
        unsigned int n;
        while(scanf("%u", &n) == 1 && n!=0){
     printf("1/2\n");
        }
        return 0;
}
```

46.UVA_12502_Asm

```
include 'emu8086.inc'
.model small
.data
   t dw ?
   a dw ?
   ans1 dw ?
   b dw ?
   c dw ?
.code
main proc
   call scan_num
   printn
   mov t, cx
while:
   mov ax, t
   cmp ax, 0
   je exit
   call scan_num  ; cin>a
   printn
   mov a, cx
   call scan_num    ;cin>b
   printn
   mov b, cx
   call scan_num  ;cin>c
   printn
   mov c, cx
            ;c*(2*a*b)/(a+b)
   mov ax, a
   mov cx, 2
   mul cx     ;cx=a*2
   sub ax, b  ;cx-=b
   mov cx, c
   mul cx     ;ax=c*cx

   xor dx, dx ;dx=0
   mov bx, a  ;bx=a
   add bx, b  ;bx+=b
   div bx     ;ax=ax/bx
            ;displaying result
   call print_num
   printn

   dec t
 jmp while

exit:
   mov ah, 4ch
   int 21h
   main endp
define_scan_num
define_print_num
define_print_num_uns
end main
```

46.UVA_12502_C

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   long long t,a,b,c;
   cin>>t;
   while(t--)
   {
      cin>>a>>b>>c;
      cout<<c*(2*a-b)/(a+b)<<endl;
   }
   return 0;
}
```

47.UVA_12577_Asm

```
include 'emu8086.inc'
.model small
.stack 100h
.data
  str db 100 dup(?)
  char db ?
.code
main proc


while:
   mov bx, 0
 input:
   mov ah, 1      ; input char
   int 21h
   mov char, al   ; char=al
   cmp al, 2Ah    ; if * then exit
   je exit
   mov al, char   ; char=newline then exit Input
   cmp al, 13
   je exit_input
   mov al, char
   mov str[bx], al ; str[bx]=char
   inc bx        ; bx++
   jmp input
exit_input:
   printn
   mov si, 0     ;si=0
   cmp str[si], 48H  ;48H='H'
   je akbar
   printn "Hajj-e-Asghar"
   jmp after
akbar:
   printn "Hajj-e-Akbar"

after:            ; jump while
jmp while

 exit:
   mov ah, 4ch
   int 21h
   main endp
end main
```

47.UVA_12577_C

```
#include<bits/stdc++.h>

#define LL long long

using namespace std;

int main()
{
   string s;
   int i = 1;
   while(cin >> s && s != "*")
   {
      cout << "Case "<< i<<": ";
      if(s[0] == 'H')
      {
         cout << "Hajj-e-Akbar\n";
      }
      else
      {
         cout << "Hajj-e-Asghar\n";
      }
      i++;
   }
   return 0;
}
```

48.UVA_13012_Asm

```
INCLUDE 'emu8086.inc'  ; include library
function
.MODEL SMALL
.STACK 100H
.DATA            ; data section
  c dw ?
  i dw ?
  n dw ?
  t dw ?
.CODE            ; code section
MAIN PROC        ; main proc start
   MOV AX, @DATA ; import data
   MOV DS, AX
while:
  call scan_num; input t
  mov t, cx
  printn
  mov i, 1                  ;i=1
  mov n, 0                          ;n=0
 for:
  mov ax, i
  cmp ax, 5
  jg exit_for

  call scan_num
  mov c, cx
  printn

  mov cx, c       ;compare c and t
  cmp cx, t
  je nInc
  jmp for_jump

 nInc:
   inc n
for_jump:
  inc i
  jmp for
 exit_for:
 mov ax, n
 call print_num
 printn

   jmp while
Exit:
   MOV AH, 4CH   ; exit
   INT 21H
   MAIN ENDP      ; main proc end
   DEFINE_PRINT_NUM_UNS
   DEFINE_PRINT_NUM
   DEFINE_SCAN_NUM
END MAIN
```

48.UVA_13012_C

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   int t,c,i,n;
   while(cin>>t)
   {
      n=0;
   for(i=1;i<=5;i++)
   {
      cin>>c;
      if(c==t)
      {
         n++;
      }
   }
   cout<<n<<endl;
   }
   return 0;

}
```

49.UVA_13025_Asm

```
INCLUDE 'emu8086.inc'  ; include library function
.MODEL SMALL
.STACK 100H
.DATA            ; data section
.CODE            ; code section
MAIN PROC        ; main proc start
                 ; diplaying result
   printn "May 29, 2013 Wednesday"

Exit:
   MOV AH, 4CH   ; exit
   INT 21H
   MAIN ENDP     ; main proc end
END MAIN
```

49.UVA_13025_C

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   cout<<"May 29, 2013 Wednesday"<<endl;
   return 0;
}
```

50.UVA_10812_Asm

```
include 'emu8086.inc'  ; include library function
.model small
.stack 100h
.data            ; data section
  T dw ?
  s dw ?
  d dw ?
  x dw ?
  y dw ?
  i dw ?

.code            ; code section
main proc        ; main proc start
   mov ax, @data ; import data
   mov ds, ax

 while:
     call scan_num    ; input T
     mov T, cx
     printn
      mov i, 0     ; i=0

 for:
   mov ax, i      ;ax=i
   cmp ax, T      ;ax=T
   je exit_for

   call scan_num  ;input s
   mov s, cx      ;s=cx
   printn
   call scan_num   ;input d
   mov d, cx
   printn        ;newline

 mov ax, s       ;ax=d
 cmp ax, d       ;compare s and d, s<d
 jg print_impossible

 xor dx, dx     ;dx=0
 mov ax, s      ;(s-d)%2!=0
 sub ax, d
 mov cx, 2
 div cx
 cmp dx, 0
 jne print_impossible
 jmp else

print_impossible:
  printn "impossible" ;displaying impossible
  jmp after
else:
   xor dx, dx
   mov ax, s     ;ax=s
   sub ax, d     ;ax=s-d
   mov cx, 2     ;c=2
   div cx        ;ax=ax/cx
   mov y, ax     ;y=ax

   mov ax, s     ;ax=s
   add ax, d     ;ax+=d
   mov cx, 2
   div cx
   mov x, ax

   mov ax, x     ;displaying x
   call print_num
   print " "      ;displaying space
   mov ax, y
   call print_num  ;displaying y
   printn
after:
```

```
    inc i ;i++      ;i++
  jmp for  ; jump to for

exit_for:
 jmp while  ;jump to while

exit:
    mov ah, 4ch   ; exit
    int 21h
    main endp      ; main proc end
                ;define build in function
    define_print_num_uns
    define_print_num
    define_scan_num
end main
```

50.UVA_10812_c

```
#include<stdio.h>

int main()
{
    int T, s, d, x, y, i;
    while(scanf("%d", &T)==1)
    {
        for(i=1; i<=T; i++)
        {
            scanf("%d%d", &s, &d);
            if(s<d || (s-d)%2!=0)
            {
                printf("impossible\n");
            }
            else {
                y = (s-d)/2;
                x = (s+d)/2;
                printf("%d %d\n", x, y);
            }

        }
    }
    return 0;
}
```