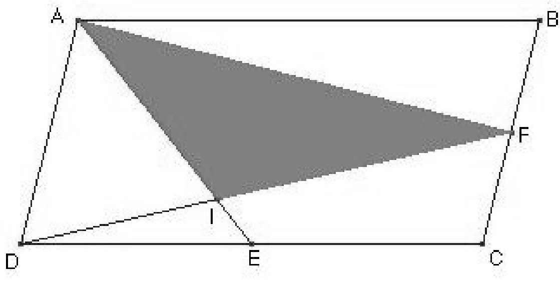
Problème du trimestre n°21

Soit un parallélogramme ABCD, E le milieu de [DC] et F le milieu de [BC] ; (AE) et (DF) se coupent en I.
Monter que l'aire du triangle AIF vaut 30 % de l'aire du parallélogramme.

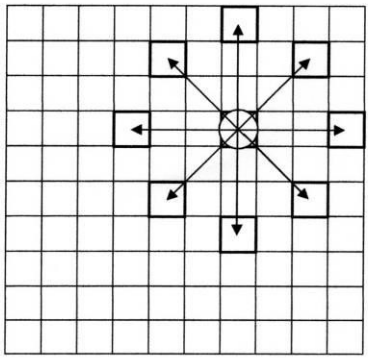


Ce mois-ci, le problème proposé est relativement facile. Mais nous aimerions avoir le maximum de réponses, utilisant les « cadres » les plus divers.

SOLUTION DU PROBLÈME N° 20

Au zoo de Raon-l'Étape vit l'ami ELTON, un kangourou solitaire enfermé dans un enclos de 10 m sur 10 m pavé par 100 dalle de ciment. Pour tromper son ennui, l'ami Elton saute de case en case :

- soit de 3 m vers le nord, l'est, le sud ou l'ouest,
- soit de $2\sqrt{2}$ m vers le nord-est, le sud-est, le sud-ouest ou le nord-ouest.



Inlassablement, en 100 sauts, il revient à la case départ en ayant piétiné toutes les cases.

Sauriez-vous retrouver un des chemins de l'ami Elton ?

Nous avons reçu une solution (partielle) de François **MUNIER** de LUSSE (Vosges). Pas la peine de se fatiguer... c'est l'ordinateur qui va faire le travail. Voici son programme en Pascal :

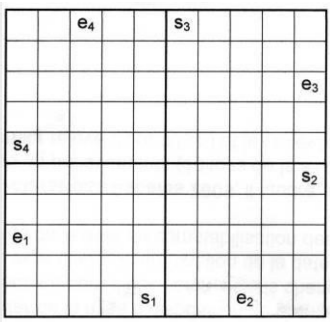
```
program pb_20 ;
const ncolo=5 ; nlign=5 ; xdep=1 ; xarr=5 ; ydep=3 ;
yarr=1 ;
type t=array[1...nlign,1...ncolo] of integer ;
var enclos :t ; i,j :integer ;

procedure  deplace(x,y,n :  integer ;  table :  t) ;  var
tableloc : t ;
   procedure essai (a,b : integer) ;
      function possible(abs,ord : integer) : boolean ;
      begin
         possible:=((abs>1)and(abs<ncolo)and(ord>1)
         and(ord<nlign))and((table[nlign-ord+1,abs]=-1)
         or(table[nlign-ord+1,abs]=9999)and(n=2))) ;
      end ;
   begin (* début essai *)
   if possible(a,b) then if n=2 then begin
              for i:=1 to nlign
      do begin for j:=1 to ncolo
              do write(table[i,j],' ',) ;
              writeln ; end ;
              writeln ;
              end
      else begin
              tableloc:=table ;
              tableloc[nlign-b+1 ;a]:=nlign*ncolo-n+1 ;
              deplace(a,b,pred(n),tableloc) ;
              end ;
   end ;
begin (* debut procédure deplace *)
   essai(x,y-3) ; essai(x,y+3) ; essai(x-3 ;y) ;
   essai(x+3 ;y) ; essai(x-2,y-2) ; essai(x-2,y+2) ;
   essai(x+2,y-2) ; essai(x+2,y+2) ;
end ;

begin (* début programme principal *)
   for i:=1 to nlign do
      for j:=1 to ncolo do enclos[i,j]:=-1 ;
   enclos[nlign-ydep+1,xdep]:=0 ;
   enclos[nlign-yarr+1,xarr] :=9999 ;
   deplace(xdep,ydep,ncolo*nlign,enclos) ;
end.
```

Premiers essai avec `Xdep = Xarr = 1` et `Ydep = Yarr = 1` (on fait partir l'ami Elton du coin nord-ouest de son enclos), et en remplaçant `n:=2` par `n:=1` dans la procédure `essai` et dans la fonction `possible`.

Il n'y a aucune solution pour un carré de 4×4 (réponse instantanée de l'ordinateur).
Pour un carré 5×5, l'ordinateur donne toutes les solutions en quelques secondes.
Pour un carré 6 ×6, la première réponse arrive après quelques heures de travail (*N.d.l.r. 2010 : on est en 1989, il y a eu des progrès depuis...*)... inutile, par conséquent, d'essayer avec le carré 10×10 !
Il faut donc aider la machine : par exemple en partageant en quatre carrés 5 ×5 avec des points d'entrée et de sortie bien choisis, comme ceci par exemple :



Voici une des solutions obtenues avec cette méthode :

92	97	76	91	98	75	58	68	74	57
78	89	94	79	88	66	61	71	54	62
84	81	99	85	82	69	52	64	59	51
93	96	77	90	95	72	55	67	73	56
100	86	83	80	87	65	60	70	53	63
23	13	7	20	12	29	47	44	41	50
16	4	10	15	5	34	39	27	33	36
1	19	24	2	8	45	42	30	46	43
22	14	6	21	11	28	48	35	40	49
17	3	9	18	25	31	38	26	32	37