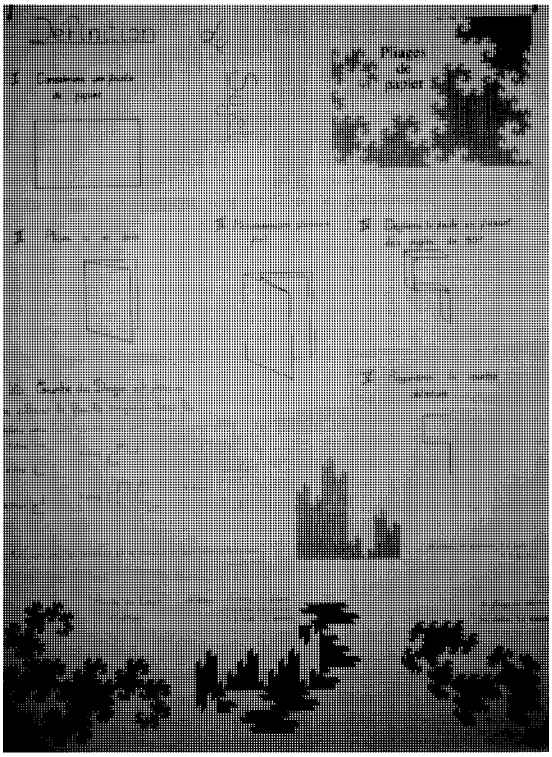
le pliage de papier

par Aurélie Desènne (2nde), Benoît Mariette (1ère) et Adeline Roux (Tle), du lycée Jean Jaurès d'Argenteuil (95)

enseignants : Joseph Cesaro, René Veillet

chercheurs : Daniel Barsky et François Digne



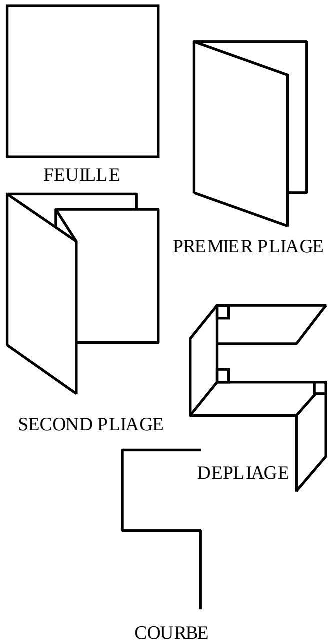
Compte-rendu de l'exposé par les parrains du groupe : **lycée A. Kastler**

— Présentation intéressante / explications suffisantes pour comprendre tout de suite les démonstrations / dessins formés très chouettes

— nous avons été étonnées de découvrir toutes les figures obtenues à partir de pliages de papier. Bravo à ceux qui se sont investis dans ce sujet !!

présentation du sujet

Plions une feuille de papier plusieurs fois en rabattant soit le côté gauche sur le côté droit, soit le côté droit sur le côté gauche. Déplions-la maintenant en formant à la place des plis des angles droits. Nous obtenons la courbe ainsi formée sur la tranche de la feuille.

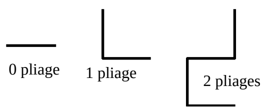


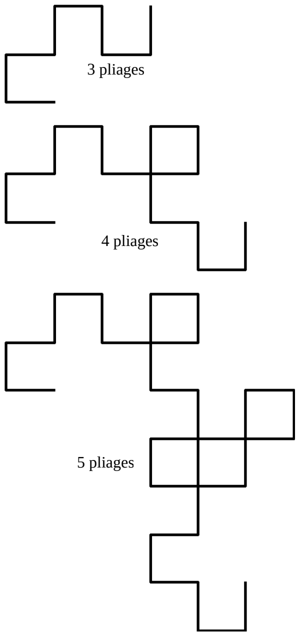
Nous présenterons nos recherches en commençant par vous montrer la progression d'une courbe en fonction du nombre de pliages, de là nous pourrons immédiatement tirer des généralités sur toutes les courbes et nous finirons en vous exposant les particularités de certaines courbes que nous avons observées plus en détail.

génération d'une courbe

la courbe du dragon par exemple

La courbe du dragon est la courbe obtenue si l'on plie la feuille toujours de la droite vers la gauche ou toujours de la gauche vers la droite :





généralités

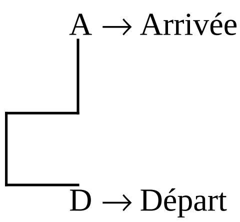
Rien qu'en observant la progression d'une telle courbe, on tire déjà plusieurs propriétés de n'importe quelle courbe, propriétés que vous avez sans doute vous-même remarquées :

la courbe ne se recoupe jamais

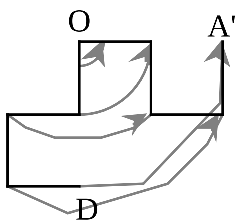
En effet, et quelque soit le nombre de pliages que l'on fait la courbe se rejoint en certains endroits mais jamais ne se coupe. En fait, cette propriété est évidente car la courbe a la forme de la tranche d'une feuille, laquelle feuille ne peut bien sûr pas se couper.

[NDLR : pas si évident ... si on tient à déplier systématiquement à 90°, est-ce qu'on arrive à le faire sans couper la feuille ?]

une simple rotation différencie les courbes de n pliages et de (n+1) pliages



2 pliages



3 pliages

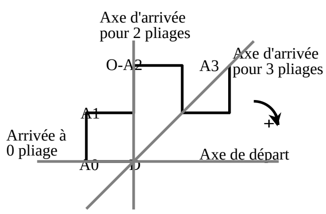
On voit bien sur ces courbes que l'on retrouve l'intégralité de la courbe à 2 pliages sur la courbe à trois pliages de D à O et que la portion de courbe de O à A' n'est que l'image, par la rotation de centre O et d'angle 90°, de la portion de la courbe DO.

Ceci s'explique encore tout bêtement : si l'on choisit de ne déplier que 2 pliages de la feuille sur laquelle on en a effectué 3, on obtiendra alors la courbe à 2 pliages, si ensuite on déplie le troisième pliage, on retrouvera donc la courbe à 2 pliages de part et d'autre du troisième pliage, ce dernier étant ouvert à 90°, la rotation est évidente.

De cette propriété découlent 2 autres évidentes :

l'angle axe de départ/axe d'arrivée

Ce que nous appelons *axe de départ*, c'est la droite qui porte le segment qu'est la courbe à 0 pliage ; ce que nous appelons *axe d'arrivée*, c'est la droite qui relie le point de départ au point d'arrivée.



D'après la propriété précédemment énoncée, A_3 est l'image de D par $R(O\,;-90°)$ soit

$$(OD, OA_3) = -90° \text{ et } OD = OA_3.$$

On a donc (O, D, A_3) triangle isocèle rectangle en O et l'angle $(DO, DA_3) = 45°$. Or le point O est aussi le point A_2, l'axe d'arrivée a donc effectué une rotation d'angle 45°.

Si on reprend l'exemple de la courbe du dragon, à chaque pliage de plus, l'axe des arrivées effectue une rotation de 45° dans le sens positif soit l'axe d'arrivée au bout de n pliages fait un angle de n.45° avec l'axe de départ.

En généralisant, un pliage à droite entraîne une rotation positive et un pliage à gauche, une rotation négative, de telle sorte que pour une courbe à n pliages à gauche et m pliages à droite, l'angle entre les deux axes vaut

$$(m - n).45°.$$

distance du point d'arrivée au point de départ

Nous avons vu précédemment que le triangle (A_n, D, A_{n+1}) était isocèle rectangle en A_n ; si l'on considère la distance DA_{n+1} par rapport à DA_n, on obtient $DA_{n+1} = \sqrt{2}.DA_n$. On prouve ainsi aisément que cette formule est récurrente, si l'on pose que le segment DA_0 vaut 1 cm, on obtient la formule :

$$DA_n = (\sqrt{2})^n \text{ cm}.$$

une codification facile

Tracer une courbe, cela revient finalement à aller du point D au point A en faisant des segments égaux au bout desquels on tourne à droite ou à gauche pour faire un autre segment et ainsi de suite. On peut donc coder par une suite de D et de G ou de 0 et de 1 (où par exemple le 0 voudrait dire droite et le 1 gauche).

Observons les suites obtenues pour la courbe du dragon :

1 pliage : G ou 1
2 pliages : GGD 110
3 pliages : GGDGGDD 1101100
4 pliages : GGDGGDDGGGDDGDD
ou 110110011100100

pour 4 pliages :

$$\underbrace{110\ 1100}_{\mathbf{A}}\ \mathbf{1}\ \underbrace{1100100}_{\mathbf{B}}$$

B est le miroir de **A**

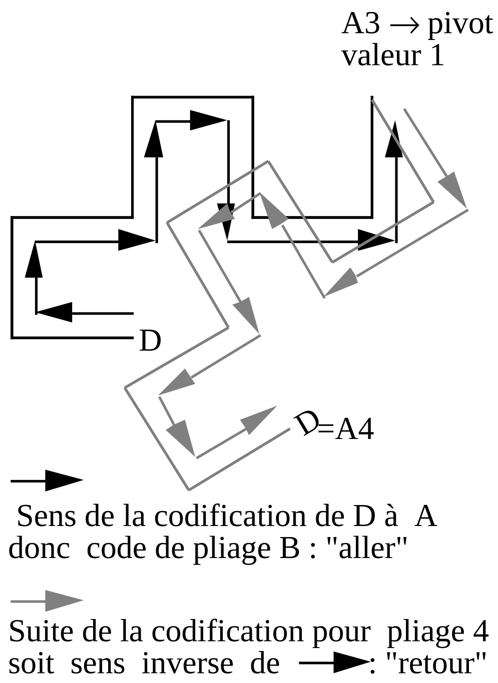
Pliage supplémentaire : valeur pivot

On observe donc que la suite est "symétrique" par rapport à la valeur pivot qui correspond au dernier pliage effectué, c'est-à-dire qu'à un 1 dans la partie **A** correspond un 0 dans la partie **B**.

Effectivement, déplier le quatrième pliage en terme de codification revient à effectuer le parcours exactement inverse (voir figure suivante) à partir du point A_3 et dans le sens que nous indique la valeur pivot.

Ainsi, si nous avons tourné à droite (0), puis à gauche (1), puis à gauche (1) à l'aller, nous tournerons à droite (0) puis à droite (0) puis à gauche (1) au retour.

Ceci étant valable dès le premier pliage, il est facile (dans le principe) de faire la codification de n'importe quelle courbe, sachant dans quel sens ont été effectués les pliages successifs (nous nommons donc les différentes courbes par ces pliages, par des D ou des G).



Cette codification ayant l'avantage de s'exprimer en binaire, nous l'avons facilement traduit en informatique et c'est ainsi que nous avons obtenu les différentes courbes présentées dans cet article. [NDLC : voir le programme donné en annexe, page 111.]

particularités de certaines courbes

En étudiant ce sujet, nous nous sommes attardés sur un de ses aspects : le remplissage du plan et le plus grand espace quadrillé que présentent les courbes.

Mais compter les petits carrés n'est pas ce qu'il y a de plus rapide ; de plus, nous avons aussi étudié des courbes qui finalement n'avaient pas d'intérêt.

Nous ne vous présenterons que la partie la plus fructueuse de nos travaux.

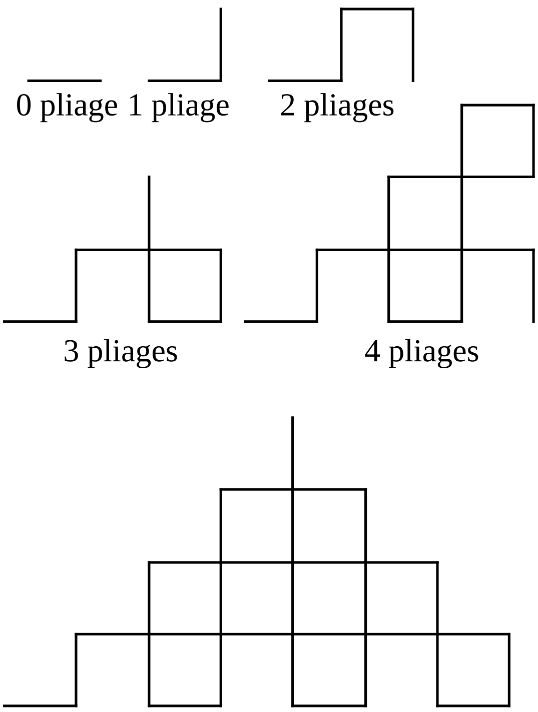
la courbe Pythagore : DGDG …

Son nom :

Nous l'avons appelée du fait de sa géométrie d'ensemble qui est toujours un triangle rectangle, et qui de plus est isocèle, quel que soit le nombre de pliages.

Son évolution :

Elle s'obtient en alternant ainsi les pliages : un à droite, un à gauche, etc.



0 pliage 1 pliage 2 pliages

3 pliages 4 pliages

5 pliages

Je suppose vraie cette affirmation :

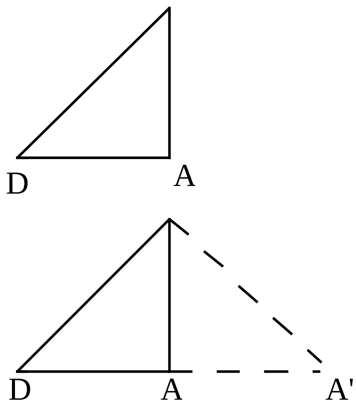
"La courbe Pythagore s'inscrit pour *n* pliages dans le 1/8 de plan qui s'ouvre à partir du départ de la courbe".

Cette affirmation est visiblement vraie pour les courbes de pliage 1 à 5.

De plus, comme nous l'avons précédemment expliqué, pour obtenir une courbe au pliage suivant, il suffit d'effectuer une rotation d'angle 90° et de centre l'arrivée de la courbe précédente.

Deux cas se présentent alors :

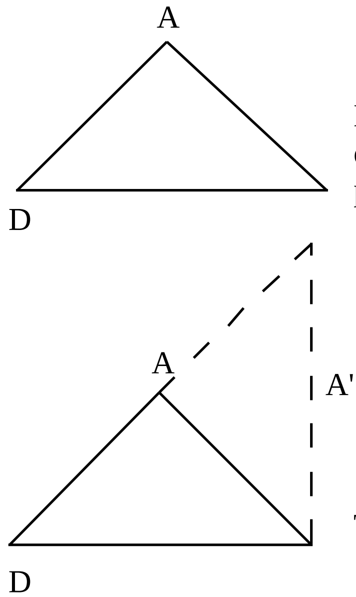
- *la courbe ressemble à :*



La rotation nous donne au $(n+1)^{\text{ième}}$ pliage :

Toujours dans le 1/8 du plan.

- *la courbe ressemble à :*



La rotation nous donne au $(n+1)^{\text{ième}}$ pliage :

Toujours dans le 1/8 du plan.

le remplissage du 1/8 de plan

Nous nous sommes amusés à compter le nombre de carrés "fermés" que contenait la courbe à n pliages. Nous avons trouvé :

nb pliages	0	1	2	3	4	5	6	7
nb carrés	0	0	0	1	3	9	21	49

La suite U_n qui représente le nombre de carrés de la courbe à n pliages est définie par :

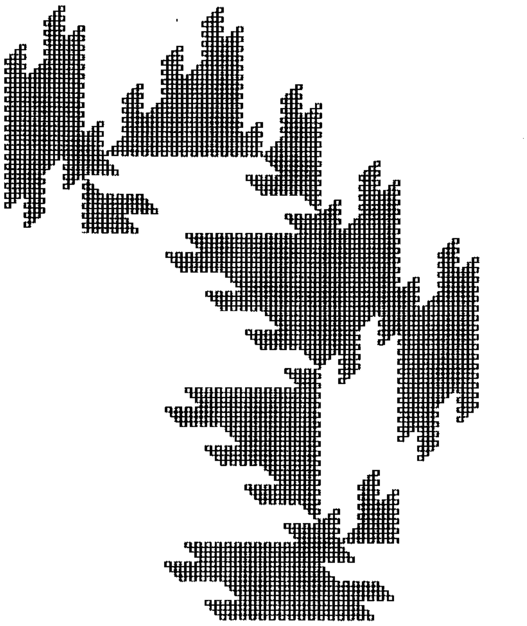
$$U_0 = 0$$
$$U_{n+1} = 2\,U_n + V_n \qquad \text{pour } n \geq 0$$

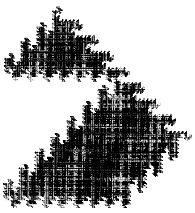
où l'on a :

$$V_0 = 0\ ;\ V_1 = 0$$
$$V_{n+1} = 2\,V_{n-1} + 1 \qquad \text{pour } n \geq 1$$

[NDLR : cette propriété complexe ne nous semble pas évidente.]

Remarquons la représentation géométrique de chaque terme de cette suite.

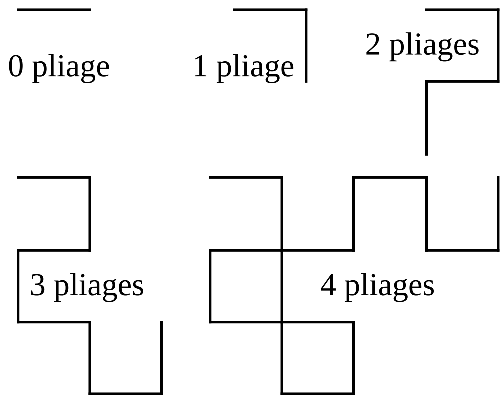


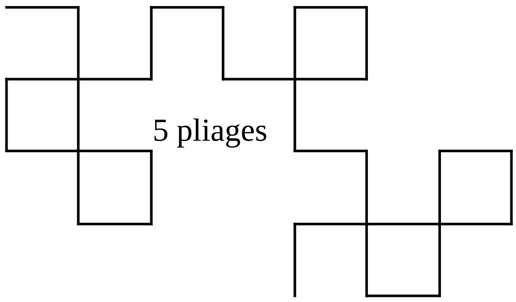


La courbe DDGG…

Son évolution

Elle s'obtient en alternant ainsi les pliages : deux à droite, deux à gauche, …





Son remplissage

On observe, sans savoir le montrer, que la courbe se cantonne à un quart de plan, toutefois, cela reste logique puisque l'axe d'arrivée balaye un quart de plan (voir définition de l'axe d'arrivée, page 106). D'autre part, on remarque que la courbe s'imbrique très bien en elle-même un pliage sur deux.

Le comptage intégral de tous les carrés ne nous amenant à rien d'évident, nous avons donc eu l'idée de ne compter que la plus grande surface rectangulaire remplie, ce qui nous donne :

nb de pliages	0	1	2	3	4	5	6	7	8	9	10
nb de carrés	0	0	0	0	1	1	6	6	42	42	210

ce qui donnerait une suite U_n, nombre de carrés fermés dans le plus grand rectangle rempli, ainsi définie :

$$U_0 = 0,$$
$$U_1 = 0,\ \ U_2 = 0,$$
$$U_3 = 0,\ \ U_4 = 1,\ \ U_5 = 1$$

$$U_{n+1} = 2\ U_{n-1}(\ 2\ U_{n-1} + 1) \quad \text{pour } n \geq 6$$

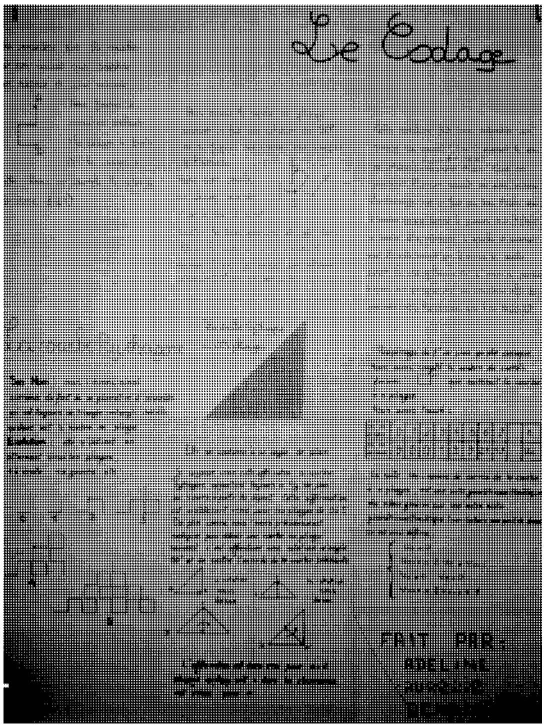
Mais cette suite reste “indénommée” … [NDLC : c’est-à-dire cette formule n’est pas démontrée.]

conclusion

C’est certainement bien mais d’autres courbes présentent sans doute d’autres particularités, je vous donne pour exemple :

DGDDGGDDDGGG… *DDDGGG…*

Merci de vous être arrêté sur notre article.



[NDLR : voir aussi l’article suivant, page 113.]

```
program dragon;
uses graph,crt;

const
  {h=2;}
  mp=15;
  mt=32768;

  type
    ind1=0..2;
    ind2=1..mt;

  var
    h,x,x1,x2,mx,y,y1,y2,my:integer;
    s:string[mp];
    er:boolean;
    drd:char;
    dir,r:string[1];
    i,j,fintab:ind2;
    tab:array[ind2] of ind1;
  {-------------------------------------------------
  procedure graphinit;
  var
    graphpilote, graphmode,coderreur:integer;
    chemin:string[20];

  begin
    graphpilote:=detect;
    chemin:='c:\prog\tp6\bgi';
    initgraph(graphpilote,graphmode,chemin);
    coderreur:=graphresult;
    if coderreur<>grOk then write('erreur gra-
phique:',grapherrormsg(coderreur))
  end;
  {-------------------------------------------------
  procedure tablo;
  begin
    for i:=1 to mt do
      tab[1]:=2;
  end;
  {-------------------------------------------------
  procedure erreur(a:byte);
  begin
    closegraph;
    gotoxy(1,1);
    write('erreur',a,':program dragon termin!');
    readln;
    halt;
  end;
  {-------------------------------------------------
  procedure code;
  begin
    fintab:=1;
    while tab[fintab+1]<2 do fintab:=fintab+1;
    if (fintab*2+1)>mt then erreur(1)
    else begin
      if dir='D' then tab[fintab+1]:=1
      else begin
        if dir='G' then tab[fintab-1]:=0
        else erreur(2);
        end;
      for i:=1 to fintab do
        begin
        if tab[fintab-i+1]=1 then tab[fintab-i+1]:=0
        else tab[fintab+i+1]:=1;
        end;
    end;
  end;
  {-------------------------------------------------
  procedure trace;
  begin
    er:=false;
    mx:=getmaxX;
    my:=getmaxY;
   { x1:=mx div 2;
    y1:=my div 2;
    x2:=x1+h;
    y2:=y1;}
    moveto(x1,y1);
    line(x1,y1,x2,y2);
    fintab:=1;
    while tab[fintab+1] <2 do fintab:=fintab+1;

    for i:=1 to fintab do
      begin
      if tab[i]=1 then
        begin
        if y1=y2 then
          begin
          if x1>x2 then y2:=y2+h
          else y2:=y2-h;
          x1:=x2;
          end
        else
          begin
          if y1>y2 then x2:=x2-2*h
          else x2:=x2+2*h;
          y1:=y2;
          end;
        end;
      if tab[i]=0 then
          begin
          if y1=y2 then
            begin
            if x1>x2 then y2:=y2-h
            else y2:=y2+h;
            x1:=x2;
            end
          else
            begin
            if y1>y2 then x2:=x2+2*h
            else x2:=x2-2*h;
            y1:=y2;
            end;
          end;

    if (0<x1) and (x1<mx) and (0<x2) and (x2<mx) and
(0<y1) and (y1<my) and (0<y2) and (y2<my) then
       line(x1,y1,x2,y2)
    else er:=true;
    end;
  end;
{-------------------------------------------------
BEGIN
  clrscr;
  repeat
  tablo;

  gotoxy(1,gety+2);

  write('Quel enchanement voulez-vous? (ATTENTION
',mp,' maximum!)');
  readln(s);
  write('quel pas?');
  readln(h);
  write('quel point de dpart? valeur en x:');
  readln(x);
  x1:=x;
  write('                     valeur en y:');
  readln(y);
  y1:=y;

  write('vous partez dans quel sens?(h,b,d,g)');
  drd:=upcase(readkey);
  case drd of
    'H':begin
        x2:=x1;
        y2:=y1-h;
        end;
    'B':begin
        x2:=x1;
        y2:=y1+h;
        end;
    'D':begin
        y2:=y1;
        x2:=x1+h;
        end;
    'G':begin
        y2:=y1;
        x2:=x1-h;
        end;
  end;
  readln;
  graphinit;
  if length(s)>mp then erreur(4);
  if upcase(s[1])='D' then tab[1]:=1
  else
    begin
    if upcase(s[1])='D' then tab[1];=1
    else erreur(5);
    end;
  for j:=2 to length(s) do
    begin
    dir:=upcase(s[j]);
    code;
    end;
  closegraph;
  gotoxy(1,1);
  { for i:=1 to mt do
    write(tab[i]);  }
  graphinit;
  trace;
  readln;
  closegraph;
  gotoxy(2,2);
  if er=true then writeln('sortie d''cran');
  writeln('votre courbe: ',s);
  writeln('votre pas: ',h);
  writeln('votre point de dpart:x=',x);
  writeln('                       y=',y);
  writeln('votre direction: ',drd);
  write('voulez-vous recommencer? (o,n)');
  readln(r);
  r:=upcase(r[1]);

  until r='N';
END.
```