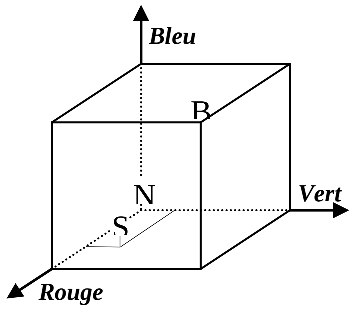
Problème 1 (Sépia contrasté)

En photographie le sépia est une qualité de tirage qui ressemble au noir et blanc, mais avec des variations de brun, et non de gris.

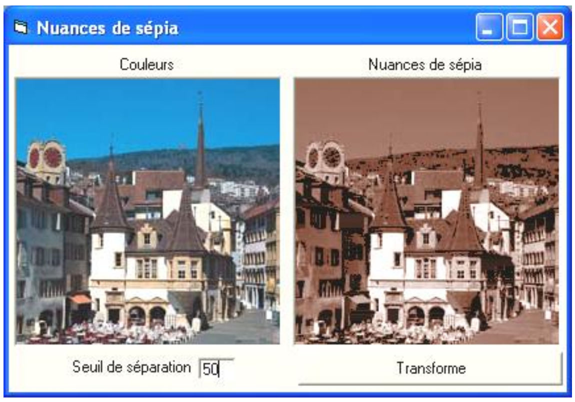
En informatique, les couleurs sont données par des intensités de rouge, vert et bleu, intensités comprises entre 0 et 255. Dans le cube des couleurs illustré ci-contre, le blanc est représenté par le point $B(255,255,255)$, le noir par le point $N(0,0,0)$ et le sépia par le point $S(94,38,18)$.



Dans la transformation d'une image couleur en une image en nuances de sépia, on tient compte d'un seuil noté s, $0 < s < 255$, qui sépare le sépia assombri et le sépia éclairci. La transformation se fait alors pixel par pixel en deux temps. Pour chaque pixel, on calcule d'abord un niveau de gris qui est la moyenne m des intensités de rouge, vert et bleu. Puis, si le gris obtenu est foncé ($m \leq s$), on le remplace par une couleur du segment NS, couleur d'autant plus proche du noir N que m est petit. Si le gris est plus clair ($m > s$), il est remplacé par une couleur du segment SB, couleur d'autant plus proche du blanc B que m est grand.

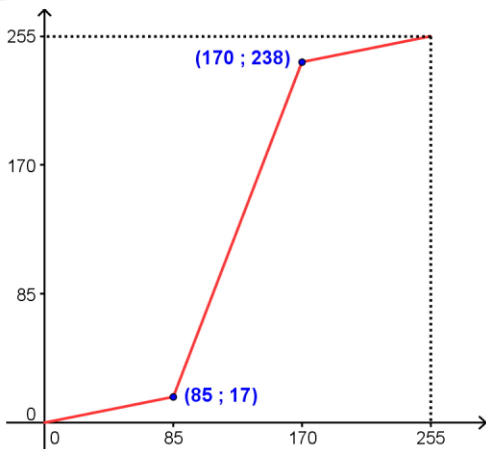
a) Un seuil étant donné dans une boîte de texte *Text*1 et une image couleur étant donnée dans une zone graphique *Picture*1, programmer la transformation de cette image en nuances de sépia et l'affichage du résultat dans une zone graphique *Picture*2.

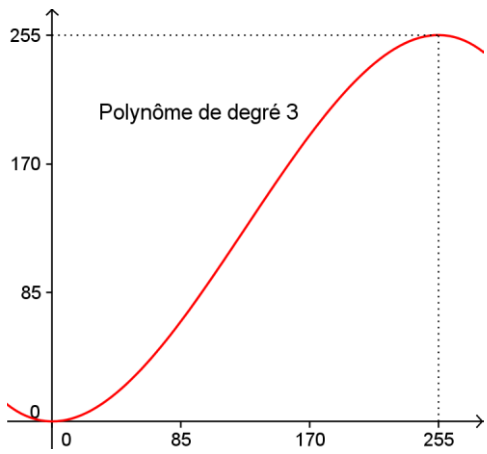
   Rappel : la couleur d'un pixel est donnée par un code $C = \text{Picture1.Point}(x, y)$. Ce code contient les intensités de rouge r, vert g et bleu b : $C = r + g \cdot 256 + b \cdot 256^2$



Pour contraster l'image en nuances de sépia, on modifie le niveau de gris m de manière à éclaircir les gris pâles et à assombrir les gris foncés. Cette modification est donnée par une fonction $m \mapsto f(m)$; elle intervient avant la conversion du gris en sépia.

b) Écrire en VisualBasic, l'expression de chacune des fonctions de contraste données par les graphes ci-dessous.





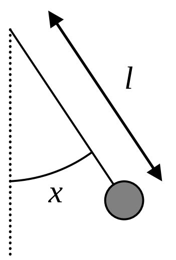
c) Proposer graphiquement une fonction qui permettrait de diminuer le contraste.

Problème 2 (Simulation du botafumeiro)

Le botafumeiro est un énorme encensoir qui se balance au bout d'une longue corde dans la cathédrale de St-Jacques de Compostelle. Un système de poulies permet à un groupe de moines de faire varier la longueur de la corde pour donner de l'élan à l'encensoir et ainsi amplifier son mouvement d'oscillation.

On se propose ici de simuler le mouvement du botafumeiro en le considérant comme un pendule de longueur variable.

Dans les calculs, on utilisera les variables et fonctions suivantes.



Au temps t exprimé en secondes, on note

$l = l(t)$	la longueur de la corde
$x = x(t)$	l'angle orienté que font la corde et la verticale ; cet angle est mesuré en radians
$x' = x'(t)$	la vitesse angulaire du pendule en radians/seconde
$x'' = x''(t)$	l'accélération angulaire du pendule

Le mouvement du pendule est régi par la loi de Newton

$$\boxed{x'' = -\frac{k}{l(t)} \cdot \operatorname{sgn}(x') \cdot (x')^2 - 2\frac{l'(t)}{l(t)} \cdot x' - \frac{g}{l(t)} \cdot \sin(x)}$$

Dans cette équation, on a un coefficient de frottement $k = 2$, la constante de l'accélération de la pesanteur terrestre $g = 10$ et la longueur variable de la corde $l(t) = 20 + 2\sin(\omega \cdot t)$.

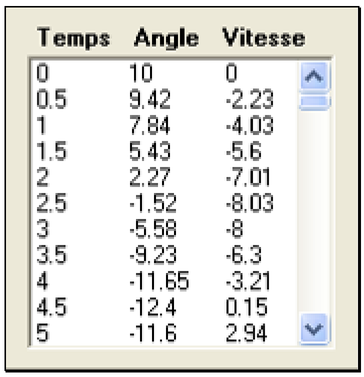
La fréquence ω à laquelle on tire et relâche la corde est décisive pour l'amplification du mouvement. On choisit ici ω=1.33

Au départ du pendule, en $t = 0$, l'angle vaut $x = 10^\circ = \dfrac{\pi}{18}$ et la vitesse angulaire est nulle.

a) Après avoir vérifié que l'accélération en $t = 0$ vaut $x'' = -\frac{1}{2}\sin(\frac{\pi}{18})$, estimer

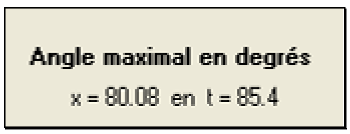
- la vitesse x' en $t = \frac{1}{4}$ en utilisant la méthode d'Euler avec un seul pas,
- l'angle x en $t = \frac{1}{2}$ en utilisant la méthode de Runge avec un seul pas.

b) En utilisant un pas temporel $h = 1/128$, écrire un programme qui estime de proche en proche, par la méthode de Runge, l'angle x et la vitesse angulaire x' durant 2 minutes, c'est-à-dire jusqu'au temps $t = 120$. Comme le montre la figure ci-contre, le programme affichera toutes les demi-secondes le temps en secondes, l'angle converti en degrés et la vitesse angulaire convertie en degrés par seconde. Les résultats convertis sont arrondis au centième.

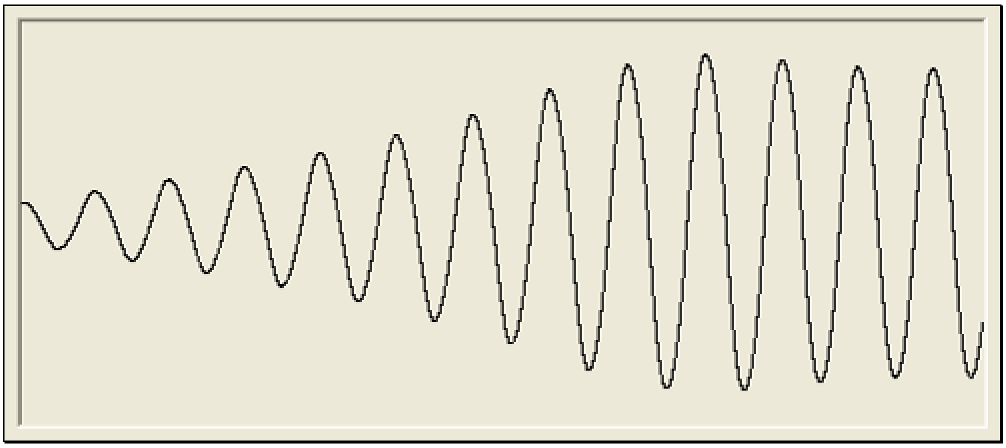


Temps	Angle	Vitesse
0	10	0
0.5	9.42	-2.23
1	7.84	-4.03
1.5	5.43	-5.6
2	2.27	-7.01
2.5	-1.52	-8.03
3	-5.58	-8
3.5	-9.23	-6.3
4	-11.65	-3.21
4.5	-12.4	0.15
5	-11.6	2.94

c) On veut connaître le maximum de la valeur absolue de l'angle. En tenant compte de toutes les valeurs calculées à la partie b), compléter le programme pour qu'il affiche une estimation de cette valeur et du temps en lequel elle a été obtenue.



d) On veut tracer le graphe qui donne l'angle en fonction du temps durant les 120 premières secondes. Pour cela, on dispose d'une zone graphique *Picture*1 comportant des pixels numérotés de 0 à 480 horizontalement et de 0 à 200 verticalement. Sachant que les angles sont tous compris en -90^{o} et $+90^{o}$, compléter le programme pour qu'il dessine le graphe.



e) Sur le graphe précédent, on voit des points à tangente horizontale.
Compléter le programme pour qu'il estime les coordonnées de ces points et les affiche dans une nouvelle liste.

Temps	Angle
0	10
4.48	-12.4
9.04	15.07
13.68	-17.87
18.34	20.78
23.03	-23.78
27.74	26.91
32.47	-30.22
37.21	33.78
41.96	-37.65
46.73	41.91

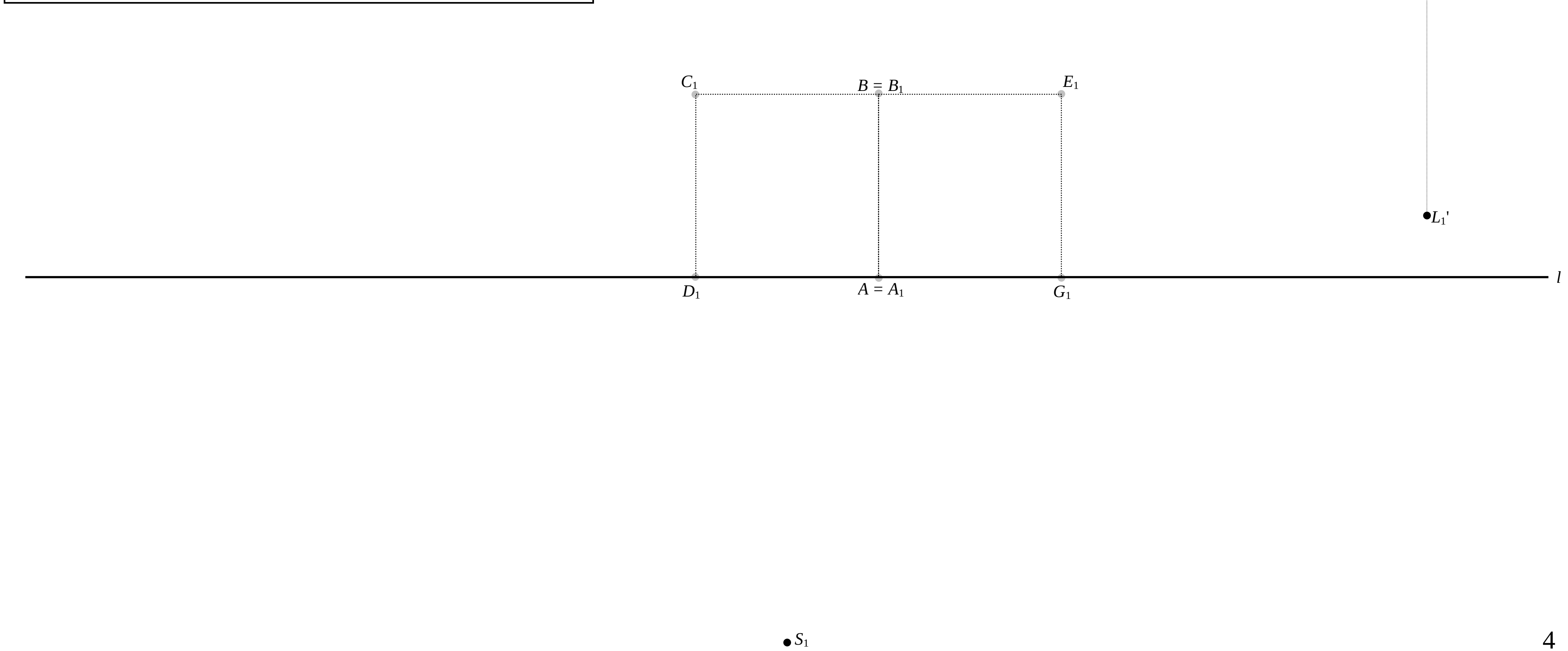
Problème 3 (Perspective)

Nom : ____________________ **Classe :** ____________________

Construire, sur l'écran donné par l et à partir du point S de cote 9 cm, la perspective d'une lettre **V** formée de 2 rectangles $ABCD$ et $ABEG$ de 3cm x 5cm, d'épaisseur négligeable et donnés par leur projection sur le sol. Les points A et B sont sur le sol.

Cette lettre est éclairée par une source lumineuse ponctuelle L donnée par sa perspective L' et par la perspective L_1' de la projection de L sur le sol. Construire la perspective de l'ombre du **V** sur le sol et sur lui-même.

Construire la projection L_1 de la source lumineuse sur le sol et donner la cote du point L.

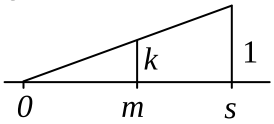


Solution 1 (Sépia contrasté)

a) Nouvelle couleur si $m \le s$

$\overrightarrow{OC} = \overrightarrow{ON} + k\,\overrightarrow{NS}$ avec

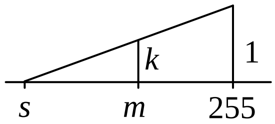
$\frac{1}{s} = \frac{k}{m} \Rightarrow k = \frac{m}{s}$



Nouvelle couleur si $m > s$

$\overrightarrow{OC} = \overrightarrow{OS} + k\,\overrightarrow{SB}$ avec

$\frac{1}{255 - s} = \frac{k}{m - s} \Rightarrow k = \ldots$



Programme

```
Let rs = 94
Let gs = 38
Let bs = 18
Let s = 0 + Text1.Text ' séparation sépia noirci, sépia éclairci
Let xmax = Picture1.ScaleWidth - 1
Let ymax = Picture1.ScaleHeight - 1
For y = 0 To ymax
  For x = 0 To xmax
    Let c = Picture1.Point(x, y)
    Let b = Int(c / 256 ^ 2)
    Let g = Int(c / 256 - b * 256)
    Let r = c - g * 256 - b * 256 ^ 2
    Let m = (r + g + b) / 3 ' moyenne
    If m <= s Then
      'OC = ON + k * NS, k=0 si m=0, k=1 si m=s
      Let k = m / s
      Let c = RGB(k * rs, k * gs, k * bs)
    Else
      ' OC = OS + k * SB, k=0 si m=s, k=1 si m=255
      Let k = (m - s) / (255 - s)
      Let c = RGB(rs + k*(255 - rs), gs + k*(255 - gs), bs + k*(255 - bs))
    End If
    Picture2.PSet (x, y), c
  Next x
Next y
```

b) Expressions fonctionnelles

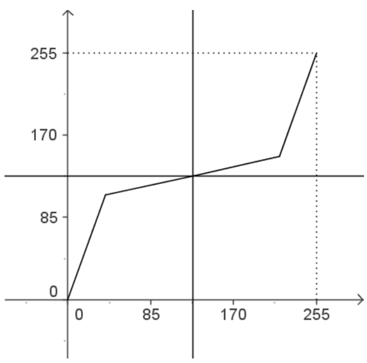
```
Function f(m)
If m < 85 Then
Let f = 17 / 85 * m
Else
  If m < 170 Then
    Let f = 221 / 85 * m - 204
  Else
    Let f = 17 / 85 * m + 204
  End If
End If
End Function
```

Polynôme $g(x) = a\,x^3 + b\,x^2 + c\,x + d$ avec

$g(0) = g'(0) = 0$, $g(255) = 255$ $g'(255) = 0$

```
Function g(x)
Let g = -2/255^2 * x^3 + 3/255 * x^2
End Function
```

c) Exemple



Solution 2 (Simulation du botafumeiro)

a) En $t=0$, on a : $l(0)=20$ et $x''=-\frac{10}{l(0)}\cdot\sin(\frac{\pi}{18})=-\frac{1}{2}\sin(\frac{\pi}{18})$

- la vitesse x' en $t=\frac{1}{4}$ en utilisant la méthode d'Euler avec un seul pas de taille $h=\frac{1}{4}$

  gauche : $t_0=0$, $x_0=\frac{\pi}{18}$, $x'_0=0$, $x''_0=-\frac{1}{2}\sin(\frac{\pi}{18})$; on suit la tangente

  droite : $t_1=t_0+h=\frac{1}{4}$, $x'_1=x'_0+h\cdot x''_0=-\frac{1}{8}\sin(\frac{\pi}{18})$

- l'angle x en $t=\frac{1}{2}$ en utilisant la méthode de Runge avec un seul pas de taille $h=\frac{1}{2}$.

  gauche : $t_0=0$, $x_0=\frac{\pi}{18}$, $x'_0=0$, $x''_0=-\frac{1}{2}\sin(\frac{\pi}{18})$; on suit un demi-pas la tangente

  milieu : $t_m=t_0+\frac{h}{2}=\frac{1}{4}$, $x'_m=x'_0+\frac{h}{2}\cdot x''_0=-\frac{1}{8}\sin(\frac{\pi}{18})$; on utilise la pente au milieu

  droite : $t_1=t_0+h=\frac{1}{2}$, $x_1=x_0+h\cdot x'_m=\frac{\pi}{18}+\frac{1}{2}\cdot\left(-\frac{1}{8}\sin(\frac{\pi}{18})\right)=\frac{\pi}{18}-\frac{1}{16}\sin(\frac{\pi}{18})\cong 0.164$

b) Programme

```
' conditions initiales
Let pi = 4 * Atn(1)
Let t0 = 0
Let x0 = 10 / 180 * pi ' radians
Let xp0 = 0
List1.AddItem 0 & Chr(9) & 10 & Chr(9) & 0 ' Chr(9) ou "    "
Picture1.PSet (0, 100)
' méthode de Runge
Let h = 1 / 128
Do
  Let tm = t0 + h / 2
  Let xm = x0 + xp0 * h / 2
  Let xpm = xp0 + f(t0, x0, xp0) * h / 2
  Let t1 = t0 + h
  Let x1 = x0 + xpm * h
  Let xp1 = xp0 + f(tm, xm, xpm) * h
  ' affichage conditionnel
  If Int(t1 * 2) = 2 * t1 Then
    Let xa = Int(x1 / pi * 180 * 100 + 0.5) / 100
    Let xpa = Int(xp1 / pi * 180 * 100 + 0.5) / 100
    List1.AddItem t1 & Chr(9) & xa & Chr(9) & xpa
  End If
  ' préparation du pas suivant
  Let t0 = t1: x0 = x1: xp0 = xp1
Loop Until t1 >= 120

' fonctions
Function f(t, x, xp)
f = -2/l(t) * Sgn(xp) * xp^2 - 2 * lp(t)/l(t) * xp - 10/l(t) * Sin(x)
End Function
Function l(t)
Let l = 20 + 2 * Sin(1.33 * t)
End Function
Function lp(t)
Let lp = 0 + 2 * 1.33 * Cos(1.33 * t)
End Function
```

c) Maximum en 3 temps

- Initialisation après les conditions initiales

```
Let xmax = x0: txmax = t0
```

- Mise à jour dans la boucle avant le pas suivant

```
  If Abs(x1) > xmax Then Let xmax = x1: txmax = t1
```

- Affichage des résultats arrondis à la fin

```
Let xa = Int(xmax / pi * 180 * 100 + 0.5) / 100
Let ta = Int(txmax * 100 + 0.5) / 100
Let Label1.Caption = " x = " & xa & "  en  t = " & ta
```

d) Comme il y a beaucoup plus de temps que de pixels, il suffit de marquer un point pour chaque nouveau temps. Le changement de coordonnées est facile pour l’abscisse, et devrait tenir compte d’un changement d’orientation sur la verticale. Avant la préparation du pas suivant, on insère

```
Picture1.PSet (4 * t1, 100 - 60 * x1)
```

e) Au passage d’un sommet la dérivée change de signe, il suffit de tester ce changement juste avant le prochain pas et d’afficher la situation au milieu dans une liste

```
If Sgn(xp0) <> Sgn(xp1) Then
  Let ta = Int(tm * 100 + 0.5) / 100
  Let xa = Int(xm / pi * 180 * 100 + 0.5) / 100
  List2.AddItem ta & Chr(9) & xa
End If
```

Solution 3 (Perspective)

Nom : ____________________ **Classe :** ____________________

Ombre du V sur le sol

- $\overline{G}' = L'G' \cap L_1'G_1'$
- $\overline{E}' = L'E' \cap \overline{G}'F_p$
- $\overline{D}' = L'D' \cap L_1'D_1'$
- $\overline{C}' = L'C' \cap \overline{D}'F_p$

Enveloppe de $A'B'\overline{E}'\overline{G}'$ et $A'B'\overline{C}'\overline{D}'$

Ombre du V sur lui-même

- $\overline{X}' = A'\overline{D}' \cap \overline{G}'\overline{E}'$
- $\tilde{X}' = \overline{X}'L'$
- $\tilde{Y}' = \tilde{X}'Fp \cap B'C'$

Ombre : $A'B'\tilde{Y}'\tilde{X}'$