e3a concours

CONCOURS ENSAM - ESTP - ECRIN - ARCHIMEDE

Epreuve d'Informatique MP

durée 3 heures

Les exercices sont indépendants. Indiquez en tête de copie le nom du logiciel ou du langage que vous utilisez.

1. Écriture des fonctions sinus et cosinus

Pour calculer les valeurs des fonctions *sinus* ou *cosinus* en un point, on peut se servir de leurs développements limités. On peut montrer que le développement limité à l'ordre 3 suffit pour calculer le *sinus* ou le *cosinus* d'un argument appartenant à `[0; 0,1]` à 10^{-5} près.

Si l'argument n'appartient pas à `[0; 0,1]` on peut utiliser les formules de base de la trigonométrie pour ramener la valeur à calculer à une expression en *sinus* et *cosinus* d'arguments appartenant à `[0; 0,1]` n'utilisant que des opérations élémentaires (addition, soustraction, multiplication).

1.1. Indiquer les formules trigonométriques qui permettent d'amener les calculs de ***sin*** `x` et de ***cos*** `x` quel que soit `x` réel à des expressions en *sinus* et *cosinus* d'arguments appartenant à `[0; 0,1]`.

1.2. Écrire les procédures :

```
sinus :   paramètre      x : réel;
          résultat       s : réel;
```

qui calcule ***sin*** `x`,

et

```
cosinus : paramètre      x : réel;
          résultat       s : réel;
```

qui calcule ***cos*** `x`.

2. Qu'affichera ce programme ?

Soit la procédure suivante :

```
f : paramètre n entier;
début
  si n>=0 alors
    f(n-2);
    afficher(n);
    f(n-1)
  fin si
fin
```

Qu'affichera l'appel à `f(4)` ?

3. Élévation à une puissance entière d'une matrice carrée

3.1. Écrire une procédure permettant de calculer le produit de deux matrices carrées réelles d'ordre `n`.

3.2. Quel est, en fonction de `n`, le nombre de multiplications élémentaires (produit de deux nombres réels) effectuées lors de ce calcul ?

3.3. Quel est le nombre de multiplications élémentaires pour une élévation à la puissance `2000` d'une matrice d'ordre `10` qui multiplie la matrice par elle-même `1999` fois ?

3.4. En utilisant les relations

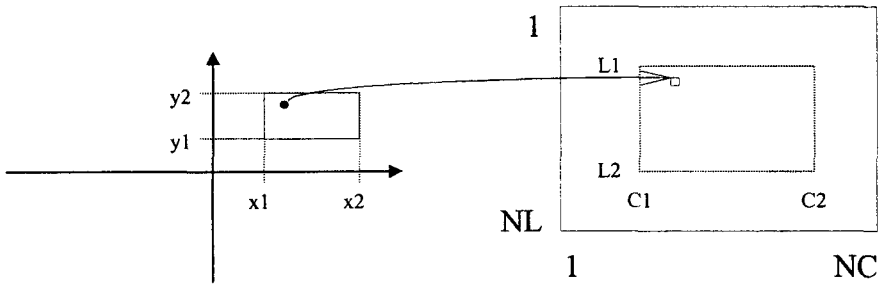
$$A^0=Id,\ A^1=A,\ A^{2k}=(A^k)^2 \text{ et } A^{2k+1}=A.A^{2k}$$

écrire une procédure récursive d'élévation à la puissance `p` d'une matrice.

3.5. Quel est, en utilisant la procédure récursive précédente, le nombre de multiplications élémentaires pour une élévation à la puissance `2000` d'une matrice d'ordre 10 ?

4. Représentation graphique sur écran

Le but de ce problème est d'écrire une procédure permettant de tracer le graphe d'une fonction réelle de variable réelle dans une fenêtre d'écran d'ordinateur associée à une portion du plan euclidien.



Plan euclidien *Écran d'ordinateur*

NL et NC sont respectivement le nombre de lignes et le nombre de colonnes de l'écran (par exemple, une définition d'écran de 800 sur 600 correspond à une valeur de 800 pour NC et une valeur de 600 pour NL).

Les représentations suivantes vous sont données à titre indicatif. Vous pouvez en utiliser d'autres à condition de les expliciter clairement.

Type de donnée	Représentation
`point_écran`	`[L,C]` où `L` et `C` sont de type `entier` et représentent la ligne et la colonne du point écran (`L` compris entre `1` et `NL`, `C` compris entre `1` et `NC`).
`fenêtre_écran`	`[CSG,CID]` où `CSG` et `CID` sont de type `point_écran` et représentent le coin supérieur gauche et le coin inférieur droit de la fenêtre.
`intervalle_réel`	`[r1,r2]` où `r1` et `r2` sont de type `réel` et représentent la borne inférieure et la borne supérieure de l'intervalle.
`portion_plan`	`[ix,iy]` où `ix` et `iy` sont de type `intervalle_réel` et représentent l'intervalle en `x` et l'intervalle en `y` de la portion rectangulaire du plan.
`point_plan`	`[x,y]` où `x` et `y` sont de type `réel` et représentent l'abscisse et l'ordonnée du point.

Attention : la portion du plan est représentée par un couple d'intervalles, alors que la fenêtre d'écran est représentée par un couple de points.

4.1. Donner les formules permettant de calculer les coordonnées d'une fenêtre proportionnelle à l'écran, centrée dans l'écran et utilisant les ¾ de sa surface.

4.2. Donner les formules permettant de calculer la position absolue `[L,C]` sur l'écran de l'image du point `[x,y]` du plan en fonction de :

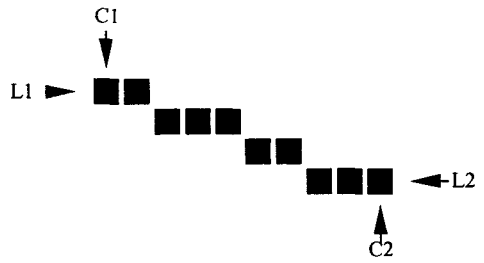
`L1`, `C1` : ligne et colonne du coin supérieur gauche de la fenêtre
`L2`, `C2` : ligne et colonne du coin inférieur droit de la fenêtre
`x1`, `x2` : les bornes en x de la portion du plan
`y1`, `y2` : les bornes en y de la portion du plan

Dans la suite du problème nous supposerons fournie la procédure

```
affiche_pixel(L,C)
```

affichant le pixel situé à la ligne `L` et à la colonne `C` de l'écran.

On désire écrire la procédure qui dessine à l'écran une représentation du segment reliant `M1[L1,C1]` et `M2[L2,C2]` de l'écran.



Comme on ne peut dessiner des points qu'aux coordonnées entières, et que les points de l'écran ont une surface, la représentation affichée aura l'allure de l'agrandissement ci-dessus.

Si `L1=L2` et `C1=C2` on affichera le pixel `[L1,C1]`.

Si `|L1-L2|>=|C1-C2|` on affichera un pixel `[L,C]` pour chaque valeur de `L` comprise entre `L1` et `L2` en déterminant de façon pertinente le `C` correspondant.

Si `|L1-L2|<|C1-C2|` on affichera un pixel `[L,C]` pour chaque valeur de `C` comprise entre `C1` et `C2` en déterminant de façon pertinente le `L` correspondant.

4.3. Écrire la procédure

```
dessine_segment_ecran : paramètres M1, M2 : point_écran;
```

qui relie les points `M1` et `M2` de l'écran.

4.4. Écrire la procédure

```
dessine_rectangle_ecran : paramètres CSG, CID : point_écran;
```

qui dessine le rectangle aux côtés parallèles aux bords de l'écran ayant `CSG` et `CID` pour coin supérieur gauche et coin inférieur droit (en utilisant `dessine_segment_ecran`).

4.5. Écrire la procédure
`dessine_rectangle_ecran2 : paramètres CSG, CID : point_écran;`
qui dessine le rectangle aux côtés parallèles aux bords de l'écran ayant `CSG` et `CID` pour coin supérieur gauche et coin inférieur droit (sans utiliser `dessine_segment_ecran`).

4.6. Écrire la procédure
```
représente_segment : paramètres    m1, m2: point_plan,
                                   p : portion_plan,
                                   F : fenêtre_écran;
```
qui trace dans la fenêtre `F` de l'écran associée à la portion de plan `p` la représentation du segment `[m1,m2]`.

4.7. Écrire la procédure
```
représente_ligne_brisee : paramètres   l : liste de point_plan,
                                       p : portion_plan,
                                       F : fenêtre_écran;
```
qui trace dans la fenêtre `F` de l'écran associée à la portion de plan `p` la représentation de la ligne brisée joignant les points de la liste `l`.

4.8. Écrire la procédure
```
représente_fonction : paramètres    f : fonction réelle,
                                    p : portion_plan,
                                    F : fenêtre_écran,
                                    n : entier;
```
qui divise l'intervalle en `x` de la portion de plan `p` en `n` parties $[x_{i-1}, x_i]$ et trace dans la fenêtre `F` de l'écran associée à la portion de plan `p` la représentation de la ligne brisée déterminée par les `n+1` points $[x_i, f(x_i)]$.