

examen d'informatique

**I fichier de texte**

Qu'affiche sur l'écran le programme suivant quand on l'exécute ?

```
var i,j,k:integer;
    c,d:char;
    s,t:string[3];
    f:text;
begin
  assign(f,'qsdf.txt'); rewrite(f);
  for i:=10 to 200 do writeln(f,i,i+2,i+4:4);
  close(f); reset(f);
  readln(f,i,j,k); writeln('i=',i,' j=',j,' k=',k);
  readln(f,s,t);   writeln('s=',s,' t=',t,'');
  readln(f,c,i,d,s,t,j);
  writeln('c=',c,' i=',i,' d=',d,' s=',s,' t=',t,' j=',j);
  close(f)
end.
```

**II matrices**

Compléter

```
const n=...;
type mat=array[1..n,1..n] of real;
operator*(a,b:mat)c:mat; ...
function puiss(a:mat;k:integer):mat; ...
```

de telle sorte que  $a*b$  soit le produit des matrices  $a$  et  $b$  et que  $\text{puiss}(a,5)$  soit égale à  $a*a*a*a*a$ .

**III Méthode de Newton**

Compléter la fonction

```
function f(a:real):real;
```

qui calcule la racine de l'équation  $x + e^x = a$ . On appliquera la méthode de Newton à la fonction  $x \mapsto x + e^x - a$ . On rappelle que  $e^x$  se note `exp(x)` en pascal.

**IV record**

Compléter la fonction

```
type point=record x,y:real end;
function milieu(a,b:point):point;
```

qui rend le milieu du segment joignant ses deux arguments.

**V fichier binaire**

Compléter le programme

```
var f:file of integer; ...
  assign(f,'A:TRUC.DAT');
  reset(f); ...
```

qui cherche où se trouve le plus grand nombre dans 'A:TRUC.DAT' et le remplace par le double du plus petit.

## Barème 7.5+4.5+3+3+5.5=24.5

0 point à une question si la réponse est identique sur deux copies.

### I fichier de texte 7.5 points

Qu'affiche sur l'écran le programme suivant quand on l'exécute ?

```
var i,j,k:integer;
    c,d:char;
    s,t:string[3];
    f:text;
begin
  assign(f,'qsdf.txt'); rewrite(f);
  for i:=10 to 200 do writeln(f,i,i+2,i+4:4);
  close(f); reset(f);
  readln(f,i,j,k); writeln('i=',i,' j=',j,' k=',k);
  readln(f,s,t);   writeln('s=',s,' t=',t);
  readln(f,c,i,d,s,t,j);
  writeln('c=',c,' i=',i,' d=',d,' s=',s,' t=',t,' j=',j);
  close(f)
end.
```

0.5 point si i=... j=... etc.

0.5 point si apostrophes bien mises s='320'

0.5 point si trois lignes

0.5 point si les valeurs contiennent bien des concaténations de nombres

11x0.5=5.5 points par valeur juste.

### II matrices 4.5 points

Compléter

```
const n=...;
type mat=array[1..n,1..n] of real;
operator*(a,b:mat)c:mat; ...
function puiss(a:mat;k:integer):mat; ...
```

de telle sorte que  $a*b$  soit le produit des matrices  $a$  et  $b$  et que  $\text{puiss}(a,5)$  soit égale à  $a*a*a*a*a$ .

opérateur \*:

0.5 point si trois boucles

0.5 point si  $s:=0$  bien placé

0.5 point si  $s+=a[i,j]*b[j,k]$

0.5 point si  $c[i,k]$  est calculé et pas de `write`

fonction `puiss`:

0.5 point si boucle for  $i:=1$  to  $n$

0.5 point si initialisation correcte ( $c:=a$ )

0.5 point si initialisation et boucle cohérentes : (for  $i:=1$  to  $n-1$ )

0.5 point si `puiss:=` et pas de `write`

0.5 point si `puiss:=` l'identité pour  $n=0$ .

### III Méthode de Newton 3 points

Compléter la fonction

```
function f(a:real):real;
```

qui calcule la racine de l'équation  $x + e^x = a$ . On appliquera la méthode de Newton à la fonction  $x \mapsto x + e^x - a$ . On rappelle que  $e^x$  se note `exp(x)` en pascal.

0.5 point si boucle `until abs(...-...)<...`

0.5 point si formule `:=x-f(x)/f'(x)`

0.5 point pour `f(x)=x+exp(x)-a`

0.5 point pour `f'(x)=1+exp(x)`

0.5 point pour initialisation correcte `x:=0` ou `x:=a` mais 0.25 pour `x:=ln(a)`

0.5 point pour `f:=x` et pas de `write`

### IV record 3 points

Compléter la fonction

```
type point=record x,y:real end;
```

```
function milieu(a,b:point):point;
```

qui rend le milieu du segment joignant ses deux arguments.

1 point pour `.x:=(a.x+b.x)/2.`

1 point pour `.y:=(a.y+b.y)/2.`

1 point pour `milieu:=...` ou `milieu.x:=...` et pas de `write`

### V fichier binaire 5.5 points

Compléter le programme

```
var f:file of integer; ...
```

```
assign(f,'A:TRUC.DAT');
```

```
reset(f); ...
```

qui cherche où se trouve le plus grand nombre dans 'A:TRUC.DAT' et le remplace par le double du plus petit.

1 point pour le parcours du fichier en lecture `while not eof(f)` ou `for i:=1 to filesize(`

1 point pour le minimum juste

1 point pour le maximum juste

1 point pour la position du maximum

1 point pour le remplacement du maximum par le double du minimum.

0.5 point pour le `close(f)`