

[X-MP-PC-2013

[> restart;

[> admet_point_fixe:=proc(t)

global n;

local i;

for i from 0 to n-1 do

if t[i]=i then

return vrai

fi;

od;

return faux;

end:

[> n:=5:t:=array(0..4,[4,3,3,1,0]):

[> admet_point_fixe(t);

faux

[> nb_points_fixes:=proc(t)

global n;

local nbre,i;

nbre:=0;

for i from 0 to n-1 do

if t[i]=i then

nbre:=nbre+1

fi;

od;

return nbre;

end:

[> t1:=array(0..4,[4,3,2,2,4]):

[> nb_points_fixes(t);nb_points_fixes(t1);

0

2

[> itere:=proc(t,x,k)

local iter,i;

iter:=x;

for i from 1 to k do

iter:=t[iter]

od;

return iter;

end:

[> seq(itere(t1,1,k),k=0..4);

1, 3, 2, 2, 2

[> nb_points_fixes_iteres:=proc(t,k)

global n;

local nbre,i;

nbre:=0;

for i from 0 to n-1 do

```

        if itere(t,i,k)=i then
            nbre:=nbre+1
        fi;
    od;
    return nbre;
end:

```

```
> seq(nb_points_fixes_iteres(t1,k),k=0..4);
```

```
5, 2, 2, 2, 2
```

```
> attire:=proc(t,i,j)
```

```

    global n;
    local j_iterre,k;
    j_iterre:=j;
    for k from 0 to n-1 do
        if j_iterre=i then
            return vrai
        else
            j_iterre:=t[j_iterre]
        fi;
    od;
    return faux;
end:

```

```
> n:=7:t:=array(0..6,[5,5,2,2,0,2,2]):
```

```
> seq(attire(t,2,j),j=0..6);
```

```
vrai, vrai, vrai, vrai, vrai, vrai, vrai
```

```
> est_attracteur:=proc(t,i)
```

```

    global n;
    local j;
    if t[i]<>i then
        return faux
    else
        for j from 0 to n-1 do
            if attire(t,i,j)=faux then
                return faux
            fi;
        od;
    fi;
    return vrai;
end:

```

```
> seq(est_attracteur(t,i),i=0..6);
```

```
faux, faux, vrai, faux, faux, faux, faux
```

```
> admet_attracteur_principal:=proc(t)
```

```

    global n;
    local i;
    for i from 0 to n-1 do
        if est_attracteur(t,i)=vrai then

```

```

        return vrai
    fi;
od;
return faux;
end:
> admet_attracteur_principal(t);
                                vrai
> temps_de_convergence:=proc(t,x)
    if t[x]=x then
        return 0
    else
        return 1+temps_de_convergence(t,t[x])
    fi;
end:
> seq(temps_de_convergence(t,x),x=0..6);
                                2, 2, 0, 1, 3, 1, 1
> allouer:=proc()
    global n;
    return array(0..n-1);
end:
> allouer();
array(0..6, [
    (0)=?_0
    (1)=?_1
    (2)=?_2
    (3)=?_3
    (4)=?_4
    (5)=?_5
    (6)=?_6
])
> temps_de_convergence_max:=proc(t)
    global n;
    local tc,i,j,k,l,ll,maxi;
    tc:=allouer();
    for i from 0 to n-1 do
        tc[i]:=n+1
    od;
    i:=0;
    while t[i]<>i do
        i:=t[i]
    od;
    tc[i]:=0;
    for i from 0 to n-1 do

```

```

    if tc[i]=n+1 then
        j:=i;
        l:=0;
        while tc[j]=n+1 do
            j:=t[j];
            l:=l+1;
        od;
        ll:=l+tc[j];
        j:=i;
        for k from 0 to l-1 do
            tc[j]:=ll-k;
            j:=t[j]
        od;
    fi;
od;
maxi:=tc[0];
for i from 1 to n-1 do
    if tc[i]>maxi then
        maxi:=tc[i]
    fi;
od;
return maxi;
end:

```

```
> temps_de_convergence_max(t);
```

3

```
> est_croissante:=proc(t)
```

```

    global n;
    local i;
    for i from 0 to n-2 do
        if t[i+1]<t[i] then
            return faux
        fi;
    od;
    return vrai;
end:

```

```
> n:=10:t:=array(0..9,[1,3,3,5,5,5,7,7,7,8]):t1:=array(0..9,[1,3,8,5,5,5,7,7,7,8]):
```

```
> est_croissante(t);est_croissante(t1);
```

vrai

faux

```
> chercher:=proc(t,a,b)
```

```

    local c;
    if a=b then
        return a
    fi;

```

```

else
  c:=floor((a+b)/2);
  if t[c]=c then
    return c
  elif t[c]>c then
    return chercher(t,c+1,b)
  else
    return chercher(t,a,c-1)
  fi;
fi;
end:

```

```
[ > chercher(t,0,9);chercher(t,0,6);
```

```
7
5
```

```
[ > point_fixe:=proc(t)
global n;
return chercher(t,0,n-1);
end:
```

```
[ > point_fixe(t);
```

```
7
```

```
[ > pgcd_points_fixes:=proc(t)
global n;
local pt;
pt:=1;
while t[pt]<>pt do
  pt:=t[pt]
od;
return pt;
end:
```

```
[ > t:=array(0..9,[0,2,4,6,4,8,0,2,0,6]):
```

```
[ > pgcd_points_fixes(t);
```

```
4
```

```
[ >
```

```
[ >
```