Mamouni My Ismail
Professeur agrégé de mathématiques
Enseignant en classes de MP
CPGE My Youssef
Rabat, Maroc
mamouni.myismail@gmail.com
myprepa.ifrance.com

ECOLE POLYTECHNIQUE
ECOLE SUPERIEURE DE PHYSIQUE ET CHIMIE INDUSTRIELLES

CONCOURS 2003 **FILIERE MP-OPTION SCIENCES INDUSTRIELLES**

FILIERE PC

EPREUVE FACULTATIVE D'INFORMATIQUE

***

l'enclos du robot

Frontière Sud-Ouest

```
STUDENT > restart;
```

Question 1.

```
STUDENT > sudouest:=proc(P,Q) local x,y;
STUDENT > x:=[P[1],Q[1]]:y:=[P[2],Q[2]]:
STUDENT > if x[1]<=x[2] and y[1]<=y[2] then 1
STUDENT > else 0
STUDENT > fi;
STUDENT > end:
STUDENT > nordouest:=proc(P,Q) local x,y;
STUDENT > x:=[P[1],Q[1]]:y:=[P[2],Q[2]]:
STUDENT > if x[1]<=x[2] and y[1]>=y[2] then 1
STUDENT > else 0
STUDENT > fi;
STUDENT > end:
STUDENT > sudest:=proc(P,Q) local x,y;
STUDENT > x:=[P[1],Q[1]]:y:=[P[2],Q[2]]:
STUDENT > if x[1]>=x[2] and y[1]<=y[2] then 1
STUDENT > else 0
STUDENT > fi;
STUDENT > end:
STUDENT > nordest:=proc(P,Q) local x,y;
STUDENT > x:=[P[1],Q[1]]:y:=[P[2],Q[2]]:
STUDENT > if x[1]>=x[2] and y[1]>=y[2] then 1
STUDENT > else 0
STUDENT > fi;
STUDENT > end:
```

Question 2

```
STUDENT > echange:=proc(a,b,i,j) local p,q,a1,b1
```

```
STUDENT > p:=min(i,j);q:=max(i,j);
STUDENT > if p=1 then
STUDENT >      if q=nops(a) then
          a1:=[a[q],seq(a[k],k=2..nops(a)-1),a[1]];
STUDENT >
          b1:=[b[q],seq(b[k],k=2..nops(b)-1),b[1]];
STUDENT >      else
          a1:=[a[q],seq(a[k],k=2..q-1),a[1],seq(a[k],k=q+1..nops(a
          ))];
STUDENT >
          b1:=[b[q],seq(b[k],k=2..q-1),b[1],seq(b[k],k=q+1..nops(b
          ))];
STUDENT >      fi;
STUDENT > else
STUDENT >      if q=nops(a) then
          a1:=[seq(a[k],k=1..p-1),a[q],seq(a[k],k=p+1..q-1),a[p]];
          b1:=[seq(b[k],i=1..p-1),b[q],seq(b[k],k=p+1..q-1),b[p]];
STUDENT >      else
STUDENT > a1:=[seq(a[k],k=1..p-1),a[q],seq(a[k],k=p+1..q-1),a[p],s
          eq(a[k],k=q+1..nops(a))];
          b1:=[seq(b[k],k=1..p-1),b[q],seq(b[k],k=p+1..q-1),b[p],s
          eq(b[k],k=q+1..nops(b))];     fi;
STUDENT > fi;
STUDENT > RETURN([a1,b1]);
STUDENT > end:
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `i` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
Warning, `k` in call to `seq` is not local
```

Question 3 *c'est le point P1*

Question 4

Ecrivons d'abord la fonction testSO qui retourne 0 si un point donné est dans la frontière SudOuest, et une valeur non nulle dans le cas contraire

```
STUDENT > testSO:=proc(a,b,i) local a1,b1,N,j;
STUDENT > a1:=op(1,echange(a,b,1,i));
STUDENT > b1:=op(2,echange(a,b,1,i));
STUDENT > N:=0:for j from 2 to nops(a) do
          N:=N+sudouest([a1[j],b1[j]],[a1[1],b1[1]]);
STUDENT > od;
```

```
STUDENT > RETURN(N);
STUDENT > end:
```

Terminons enfin par récupérer les points qui se trouvent sur la frontière SudOuest, c'est à dire pour lesquels la valeur retournée par testSO est 0

```
STUDENT > frontiereSO:=proc(a,b) local aSO,bSO,i; global nSO;
STUDENT > aSO:=NULL:
STUDENT > bSO:=NULL:
STUDENT > for i from 1  to nops(a) do
STUDENT > if testSO(a,b,i)=0 then aSO:=aSO,a[i];bSO:=bSO,b[i];
STUDENT > else fi;
STUDENT > od;
STUDENT > nSO:=nops([aSO]);
STUDENT > RETURN(seq([op(i,aSO),op(i,bSO)],i=1..nSO));
STUDENT > end:
```

Question 5

```
STUDENT > testNO:=proc(a,b,i) local a1,b1,N,j;
STUDENT > a1:=op(1,echange(a,b,1,i));
STUDENT > b1:=op(2,echange(a,b,1,i));
STUDENT > N:=0:for j from 2 to nops(a) do
          N:=N+nordouest([a1[j],b1[j]],[a1[1],b1[1]]);
STUDENT > od;
STUDENT > RETURN(N);
STUDENT > end:
STUDENT > frontiereNO:=proc(a,b) local aNO,bNO,i; global nNO;
STUDENT > aNO:=NULL:
STUDENT > bNO:=NULL:
STUDENT > for i from 1  to nops(a) do
STUDENT > if testNO(a,b,i)=0 then aNO:=aNO,a[i];bNO:=bNO,b[i];
STUDENT > else fi;
STUDENT > od;
STUDENT > nNO:=nops([aNO]);
STUDENT > RETURN(seq([op(i,aNO),op(i,bNO)],i=1..nNO));
STUDENT > end:
```

Question 6

```
STUDENT > testSE:=proc(a,b,i) local a1,b1,N,j;
STUDENT > a1:=op(1,echange(a,b,1,i));
STUDENT > b1:=op(2,echange(a,b,1,i));
STUDENT > N:=0:for j from 2 to nops(a) do
          N:=N+sudest([a1[j],b1[j]],[a1[1],b1[1]]);
STUDENT > od;
STUDENT > RETURN(N);
STUDENT > end:
STUDENT > frontiereSE:=proc(a,b) local aSE,bSE,i; global nSE;
STUDENT > aSE:=NULL:
STUDENT > bSE:=NULL:
STUDENT > for i from 1  to nops(a) do
STUDENT > if testSE(a,b,i)=0 then aSE:=aSE,a[i];bSE:=bSE,b[i];
```

```
STUDENT > else fi;
STUDENT > od;
STUDENT > nSE:=nops([aSE]);
STUDENT > RETURN(seq([aSE[i],bSE[i]],i=1..nSE));
STUDENT > end:
STUDENT > testNE:=proc(a,b,i) local a1,b1,N,j;
STUDENT > a1:=op(1,echange(a,b,1,i));
STUDENT > b1:=op(2,echange(a,b,1,i));
STUDENT > N:=0:for j from 2 to nops(a) do
          N:=N+nordest([a1[j],b1[j]],[a1[1],b1[1]]);
STUDENT > od;
STUDENT > RETURN(N);
STUDENT > end:
STUDENT > frontiereNE:=proc(a,b) local aNE,bNE,i; global nNE;
STUDENT > aNE:=NULL:
STUDENT > bNE:=NULL:
STUDENT > for i from 1  to nops(a) do
STUDENT > if testNE(a,b,i)=0 then aNE:=aNE,a[i];bNE:=bNE,b[i];
STUDENT > else fi;
STUDENT > od;
STUDENT > nNE:=nops([aNE]);
STUDENT > RETURN(seq([aNE[i],bNE[i]],i=1..nNE));
STUDENT > end:
STUDENT > frontiereSO(a,b);
```

[1, 1]

```
STUDENT > with(plots):
```

Question 7

On définit maintenant la fonction SO qui permet de tracer la frontière SudOuest en joignant ses point à l'aide de la fonction plot

```
STUDENT > tracer:=proc(P) local Pts,i;
STUDENT > Pts:=NULL:
STUDENT > for i from 1 to nops(P)-1 do
STUDENT > Pts:=Pts,P[i],[P[i][1],P[i+1][2]];
STUDENT > od;
STUDENT > Pts:=Pts,P[nops(P)];
STUDENT > end:
STUDENT > P:=[[0,11],[2,0],[2,1],[3,8],[3,7],[4,8],[4,2],[4,0],[5,
          3],[5,6],[6,9],[6,12],[6,11],[7,9],[9,3],[9,11],[10,8],[
          10,6],[10,10],[11,1]];a:=[seq(P[i][1],i=1..nops(P))];b:=
          [seq(P[i][2],i=1..nops(P))];
```

P := [[0, 11], [2, 0], [2, 1], [3, 8], [3, 7], [4, 8], [4, 2], [4, 0], [5, 3], [5, 6], [6, 9], [6, 12], [6, 11], [7, 9], [9, 3], [9, 11], [10, 8], [10, 6], [10, 10], [11, 1]]

a := [0, 2, 2, 3, 3, 4, 4, 4, 5, 5, 6, 6, 6, 7, 9, 9, 10, 10, 10, 11]

b := [11, 0, 1, 8, 7, 8, 2, 0, 3, 6, 9, 12, 11, 9, 3, 11, 8, 6, 10, 1]

```
STUDENT > P1:=[frontiereNO(a,b),frontiereNE(a,b),frontiereSE(a,b),
          frontiereSO(a,b),frontiereNO(a,b)];
```

$$P1 := [[6, 12], [6, 12], [9, 11], [10, 10], [11, 1], [4, 0], [11, 1], [2, 0], [6, 12]]$$

```
STUDENT > plot([tracer(P1)]);
```