

```

start divlimo(y);
/* divise chaque ligne par sa moyenne */
n=ncol(y);
if n=1 then ycor=y/y[:];
      else ycor=y/repeat(y[:,:],1,n);
return(ycor);
finish divlimo;

y=(repeat(1:7,4,1));
x=t(1:7);
z=1:7;
yco=divlimo(y);
xco=divlimo(x);
zco=divlimo(z);
print x xco, y yco, z zco;
w={1 2 . 3 . .,1 2 . 3 6 .};
wco=divlimo(w);
print / w, wco;

/* Stockage
libname sl 'c:\cours\sas\exos_iml';
reset storage=sl.catiml;
store module=divlimo;
show storage;
*/

/* Déstockage
libname sl 'c:\cours\sas\exos_iml';
reset storage=sl.catiml;
show storage;
load module=divlimo;
*/

start profils(plig,pcol,e);
/* calcul des profils lignes et colonnes */
p=nrow(e); q=ncol(e);
/* plig=j(p,q,0); pcol=plig; */
plig=e/repeat(e[,+],1,q);
pcol=e/repeat(e[+,],p,1);
finish profils;

e={5 2 3,0 8 2};
run profils(pl,pc,e);
print e, pl pc;

f={5 2 3,. 8 2};
run profils(plf,pcf,f);
print f, plf pcf;

```

```

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=profils;
show storage;
*/

start dnombr(ndif,dif,eff,liste);
/* compter les elements differents d'une liste */
dif=unique(liste);
ndif=ncol(dif);
eff=j(1,ndif,0);
do i=1 to ndif;
    eff[i]=sum(liste=dif[i]);
end;
finish dnombr;

liste={a b b c c b a a a b};
code={.,2,2,3,3,2,1,1,1,2};
run dnombr(nl,ldif,eff,liste);
print liste nl, ldif, eff;
z=ldif//char(eff,1); print z;
run dnombr(nc,lcod,ef,code);
print / code nc, lcod ,ef;

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=dnombr;
show storage;
*/

start clearat(dat);
/* suppression des lignes PLEINES de
manquants et/ou de zeros */
nvar=ncol(dat); nlig=nrow(dat);
verif=j(nlig,1,1);
do i=1 to nlig;
    miss=sum(dat[i,]=.);
    zero=sum(dat[i,]=0);
    if miss+zero=nvar then verific[i]=0;
end;
/* cdat=j(verif[+],nvar,.); */
cdat=dat[loc(verif),];

```

```

    return(cdat);
finish clearcat;

dat={0 0 0, . . . , 0 . 0, -1 0 1, 1 . 1};
cdat=clearcat(dat);
print dat, cdat;

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=clearcat;
show storage;
*/

start taille(x);
/*
renvoie le nombre de chiffres de x
pour x entier relatif de valeur absolue < 1e5
*/
borne={9 99 999 9999 99999};
z=x;
if z<0 then z=-z;
if z>99999 then do; chiffre=.; goto fin; end;
ok=0;
do i=1 to 5 until(ok);
if z<=borne[i] then do; chiffre=i; ok=1; end;
end;
fin:
if x>=0 then return(chiffre);
else return(chiffre+1);
finish taille;

x=-1000; chiffre=taille(x); print x chiffre;
z=-1e6; chif=taille(z); print z chif;

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=taille;
show storage;
*/

```

```

start manquant2(dat);
/* renvoie dans l'output : un message d'alerte
*/
/*
                                le nb de valeurs manquantes
*/
/*
                                et les numeros de lignes
*/
if type(dat)='N' then indice=(dat=.);
if type(dat)='C' then indice=(dat=' ');
nmiss=sum(indice);
if nmiss>0 then
    do;
        numeros=loc(indice[,+]);
        print " " /;
        if type(dat)='N' then print ">>> ATTENTION : DONNEES
MANQUANTES DE TYPE NUMERIQUE";
        if type(dat)='C' then print ">>> ATTENTION : DONNEES
MANQUANTES DE TYPE CARACTERE";
        reset noname;
        print " au nombre de :" nmiss [format=7.0];
        print " Liste des numéros de lignes :";
        print numeros;
        reset name;
    end;
    else
        print "=>> FELICITATIONS : AUCUNE DONNEE
MANQUANTE";
finish manquant2;

dat0={1 2, 3 4, 5 6};                                run
manquant2(dat0);
dat1={1 2 3 . ,1 2 3 4, 3 . 4 5,3 . . .};            run
manquant2(dat1);
dat2={1 2 3 . ,1 . . 4, 3 . 4 5,3 . . .};            run
manquant2(dat2);
dat3={'aaa', ' ', ' c ', 'd d', 'f ', ' g'}; run
manquant2(dat3);

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=manquant2;
show storage;
*/

start complet(e,code,w);
/* completion des . des lignes de e par les lignes de w
selon code */
nobs=nrow(e);

```

```

ce=e;
do i=1 to nobs;
  indice=loc(e[i,]=.);
  if ncol(indice)^=0 then ce[i,indice]=w[code[i],indice];
end;
return(ce);
finish complet;

e={. 1 .,1 3 ., . 4 .,4 2 .,2 7 9};
code={1,2,2,3,2};
w={1 2 1,2 3 1,1 1 2};
ce=complet(e,code,w);
print / e code w , ce;

/*
libname sl 'c:\cours\sas\exos_iml';
reset storage=sl.catiml;
store module=complet;
show storage;
*/

start disjbur(disj,burt,entr);
if nrow(entr)=1 then entr=entr`;
nq=ncol(entr);
disj=design(entr[,1]);
do j=2 to nq;
disj=disj||design(entr[,j]);
end;
burt=t(disj)*disj;
finish disjbur;

/* disjonctif complet pour une seule question */
q1={1, 1, 2, 3}; dq1=design(q1); print q1, dq1;

e={1 2,1 1,2 1,3 1}; run disjbur(d,b,e); print e, d, b;

/*
libname sl 'c:\cours\sas\exos_iml';
reset storage=sl.catiml;
store module=disjbur;
show storage;
*/

start tirpou1(entree,p);
/* tirer au hasard un pourcentage donné de lignes d'une
matrice */
if nrow(entree)=1 then entree=t(entree);
n=nrow(entree);

```

```

indic=j(1,n,0);
do i=1 to n;
    if uniform(0)<=p then indic[i]=i;
end;
out=entree[loc(indic),];
return(out);
finish tirpour1;

options nodate nocenter ls=120 ps=200;
entr1=repeat(char(i(5),1),20,4); print entr1;
entr2=1:100;
print /;
extr1=tirpour1(entr1,.15); print (nrow(extr1)) "sur 15",
extr1 ;
extr2=tirpour1(entr2,.2); print (nrow(extr2)) "sur 20",
extr2;

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catuml;
store module=tirpour1;
show storage;
*/

start tir(p,n);
/*
    tirage au hasard (loi uniforme)
    p nombres distincts pris dans [1,n]
    pour p<n
*/
t=j(1,p,0); k=1:n;
do i=1 to p;
    j=floor(uniform(0)*(n-i)+i); z=k[i];k[i]=k[j];k[j]=z;
end;
t=k[1:p];
return(t`);
finish tir;

ee=t('obs1':'obs15'//char(101:115));
print / ee /;
do i=1 to 2;
    tirage=tir(5,15);
    print tirage;
    print(ee[tirage,]);
end;

/*
libname sl 'c:\cours\sas\exos_uml';

```

```

reset storage=sl.catiml;
store module=tir;
show storage;
*/

goptions rotate=landscape;
proc iml;
start ar1(ro,sig,n);
/* genere et represente un ar(1) pour 0<abs(ro)<1 */
/* si ro=1 Marche aleatoire */
if ro<=-1 | ro=0 | ro>1 then
do;
  mattrib ro label="VALEUR DE RO ERRONE"; print ro;
return;
end;
y=j(n,1,.); y[1]=sig*normal(0);
do t=2 to n;
  y[t]=ro*y[t-1]+ sig*normal(0);
end;
ty=t(1:n)||y;
call pgraf(ty,'o','valeurs de t','Y','GRAPHE de (t,Y
(t))');
return(ty);
finish ar1;

z=ar1(.8,.1,200);

/*
libname sl 'c:\cours\sas\exos_uml';
reset storage=sl.catiml;
store module=ar1;
show storage;
*/

/* representation graphique 2 avec gplot */
t=z[,1]; y=z[,2];
create tabl var {t y}; append; close;
quit;
proc print; run;
symbol1 c=red v=circle i=join;
title 'GRAPHE de (t,Y(t))';
proc gplot data=tabl;
plot y*t=1; run;
title;
quit;

/* representation graphique 3 avec la procedure iml gpal
*/

```

```
libname sl 'c:\cours\sas\exos_uml';
libname sl1 'c:\cours\sas\stage';
proc uml;
reset storage=sl.catuml; load module=ar1;
reset storage=sl1.catstage; load module=gral;
ty=ar1(.8, .1, 200);
y=ty[,2];
call gstart;
run gral(y, 'AR(1): R0=.8 sig=.1 et n=200');
call gstop;
quit;
```