Algebra Lineal

Descripción General

Algebra Lineal es un conjunto de programas para la calculadora Casio Classpad orientados al tratamiento de bases de espacios vectoriales reales de dimensión finita. Implementa las principales operaciones que se pueden realizar con dichas bases: **Imagen** y **Núcleo** de una aplicación lineal, **Unión** e **Intersección** de bases, **Ortogonalización de Gram-Schmidt** y **Completar** una base así como encontrar una base del espacio vectorial **Perpendicular**.

Todas estas operaciones se dan con gran frecuencia en los ejercicios que los alumnos de las carreras técnicas (*física, química, matemáticas, arquitecturas, ingenierías...*) tienen que realizar en el transcurso de la asignatura "***Algebra Lineal***" (*común a todas ellas*).

Si bien **Algebra Lineal** está pensado como un paquete de programas relacionados, todos ellos han sido programados de manera independiente, de manera que ninguno necesita de ningún otro para funcionar pudiendo así instalar solamente los programas que prefiera el usuario.

Están pendientes de realización programas que **Diagonalicen** una matriz (o en su defecto su **Forma Reducida de Jordan**) así como los correspondientes **Cambios de Base** asociados.

Descripción de los programas

Imagen

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) correspondiente a una aplicación lineal $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) expresada en las bases canónicas, **Imagen($\mathbb{A}$)** proporciona una base del espacio vectorial imagen (que es un subespacio de $\mathbb{R}^n$) en forma de matriz $\mathbb{S}$ (*n* filas, *rank($\mathbb{A}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}1&0&3\\0&0&0\\0&0&0\end{pmatrix}\Rightarrow Imagen(A)=\begin{pmatrix}1\\0\\0\end{pmatrix}$$

$$A=\begin{pmatrix}1&1&5\\0&1&7\\0&0&0\end{pmatrix}\Rightarrow Imagen(A)=\begin{pmatrix}1&1\\0&1\\0&0\end{pmatrix}$$

$$A=\begin{pmatrix}0&0\\0&0\end{pmatrix}\Rightarrow Imagen(A)=\begin{pmatrix}0\\0\end{pmatrix}$$

Núcleo

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) correspondiente a una aplicación lineal $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) expresada en las bases canónicas, **Nucleo($\mathbb{A}$)** proporciona una base del espacio vectorial nucleo (que es un subespacio de $\mathbb{R}^m$) en forma de matriz $\mathbb{S}$ (*m* filas, *n-rank($\mathbb{A}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}1&0&3\\0&0&0\\0&0&0\end{pmatrix}\Rightarrow Nucleo(A)=\begin{pmatrix}3&0\\0&-1\\-1&0\end{pmatrix}$$

$$A=\begin{pmatrix}1&1&5\\0&1&7\\0&0&0\end{pmatrix}\Rightarrow Nucleo(A)=\begin{pmatrix}-2\\7\\-1\end{pmatrix}$$

$$A=\begin{pmatrix}0&0\\0&0\end{pmatrix}\Rightarrow Nucleo(A)=\begin{pmatrix}-1&0\\0&-1\end{pmatrix}$$

Unión

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) y una matríz $\mathbb{B}$ (*n* filas, *p* columnas) correspondientes a unas aplicaciónes lineales $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) y $\mathbb{y}=\mathbb{B}\mathbb{x}$ (de $\mathbb{R}^p$ en $\mathbb{R}^n$) respectivamente, **Union($\mathbb{A}$,$\mathbb{B}$)** proporciona una base de la unión de los espacios vectoriales imagen de ambas aplicaciones (que es un subespacio de $\mathbb{R}^n$) en forma de matriz $\mathbb{S}$ (*n* filas, *rank($\mathbb{A}$|$\mathbb{B}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}1\\0\\0\end{pmatrix};B=\begin{pmatrix}0&3\\0&0\\0&5\end{pmatrix}\Rightarrow Union(A,B)=\begin{pmatrix}1&3\\0&0\\0&5\end{pmatrix}$$

$$A=\begin{pmatrix}1&0&0\\0&0&0\\0&0&1\end{pmatrix};B=\begin{pmatrix}0&0\\0&1\\0&0\end{pmatrix}\Rightarrow Union(A,B)=\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}$$

$$A=\begin{pmatrix}0\\0\end{pmatrix};B=\begin{pmatrix}0\\0\end{pmatrix}\Rightarrow Union(A,B)=\begin{pmatrix}0\\0\end{pmatrix}$$

Intersección

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) y una matríz $\mathbb{B}$ (*n* filas, *p* columnas) correspondientes a unas aplicaciónes lineales $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) y $\mathbb{y}=\mathbb{B}\mathbb{x}$ (de $\mathbb{R}^p$ en $\mathbb{R}^n$) respectivamente, **Intersec($\mathbb{A}$,$\mathbb{B}$)** proporciona una base de la intersección de los espacios vectoriales imagen de ambas aplicaciones (que es un subespacio de $\mathbb{R}^n$) en forma de matriz $\mathbb{S}$ (*n* filas, *rank($\mathbb{A}$)+rank($\mathbb{B}$)-rank($\mathbb{A}$|$\mathbb{B}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}1\\0\\0\end{pmatrix};B=\begin{pmatrix}5&3\\0&0\\0&5\end{pmatrix}\Rightarrow Intersec(A,B)=\begin{pmatrix}5\\0\\0\end{pmatrix}$$

$$A=\begin{pmatrix}1&0\\0&0\\0&1\end{pmatrix};B=\begin{pmatrix}0\\1\\0\end{pmatrix}\Rightarrow Intersec(A,B)=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

$$A=\begin{pmatrix}1&2\\3&4\end{pmatrix};B=\begin{pmatrix}5\\10\end{pmatrix}\Rightarrow Intersec(A,B)=\begin{pmatrix}5\\10\end{pmatrix}$$

Completa

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) correspondiente a una aplicación lineal $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) expresada en las bases canónicas, **Completa($\mathbb{A}$)** proporciona una base del espacio vectorial $\mathbb{R}^n$ cuyas primeras *rank($\mathbb{A}$)* columnas forman una base del subespacio vectorial imagen de la aplicación. La salida se presenta en forma de matriz $\mathbb{S}$ (*n* filas, *rank($\mathbb{A}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}3 & 0 & 6\\ 0 & 5 & 7\\ 0 & 0 & 7\end{pmatrix}\Rightarrow Completa(A)=\begin{pmatrix}3 & 0 & 6\\ 0 & 5 & 7\\ 0 & 0 & 7\end{pmatrix}$$

$$A=\begin{pmatrix}0\\ 0\\ 0\end{pmatrix}\Rightarrow Completa(A)=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

$$A=\begin{pmatrix}0 & 0\\ 0 & 6\end{pmatrix}\Rightarrow Completa(A)=\begin{pmatrix}0 & 1\\ 6 & 0\end{pmatrix}$$

Perpendicular

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) correspondiente a una aplicación lineal $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) expresada en las bases canónicas, **Perpend($\mathbb{A}$)** proporciona una base del espacio vectorial perpendicular a la imagen de $\mathbb{A}$ (que es un subespacio de R^n complementario al de la imagen de $\mathbb{A}$). La salida se presenta en forma de matriz $\mathbb{S}$ (*n* filas, *n-rank($\mathbb{A}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}3 & 0 & 6\\ 0 & 5 & 7\\ 0 & 0 & 7\end{pmatrix}\Rightarrow Perpend(A)=\begin{pmatrix}0\\ 0\\ 0\end{pmatrix}$$

$$A=\begin{pmatrix}1\\ 0\\ 0\end{pmatrix}\Rightarrow Perpend(A)=\begin{pmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{pmatrix}$$

$$A=\begin{pmatrix}1 & 0\\ 2 & 0\end{pmatrix}\Rightarrow Perpend(A)=\begin{pmatrix}2\dfrac{\sqrt{5}}{5}\\ \dfrac{-\sqrt{5}}{5}\end{pmatrix}$$

Ortonormalización de Gram-Schmidt

Dada una matriz $\mathbb{A}$ (*n* filas, *m* columnas) correspondiente a una aplicación lineal $\mathbb{y}=\mathbb{A}\mathbb{x}$ (de $\mathbb{R}^m$ en $\mathbb{R}^n$) expresada en las bases canónicas, **ortoGS($\mathbb{A}$)** proporciona una base ortonormal del espacio imagen de la aplicación (que es un subespacio de $\mathbb{R}^n$). La salida se presenta en forma de matriz $\mathbb{S}$ (*n* filas, *rank($\mathbb{A}$)* columnas) cuyas columnas forman la citada base.

Ejemplos:

$$A=\begin{pmatrix}3 & 0 & 6\\ 0 & 5 & 7\\ 0 & 0 & 7\end{pmatrix}\Rightarrow OrtoGS(A)=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

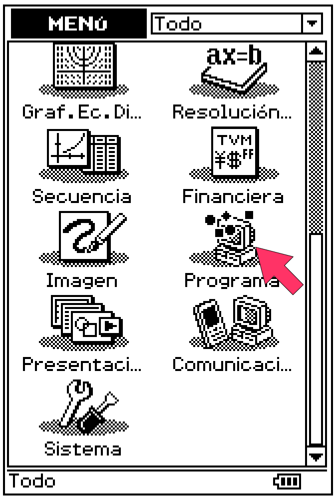
$$A=\begin{pmatrix}5\\ 0\\ 0\end{pmatrix}\Rightarrow OrtoGS(A)=\begin{pmatrix}1\\ 0\\ 0\end{pmatrix}$$

$$A=\begin{pmatrix}3 & 0\\ 4 & 0\end{pmatrix}\Rightarrow OrtoGS(A)=\begin{pmatrix}\frac{3}{5}\\ \frac{4}{5}\end{pmatrix}$$

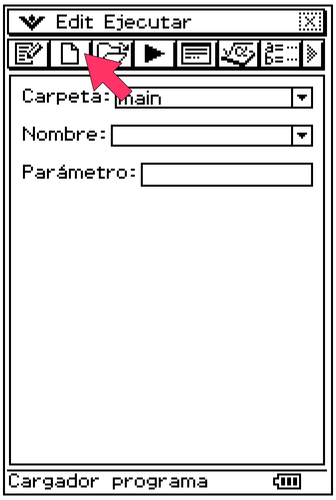
Instalación

Para instalar los programas de Algebra Lineal tan sólo tiene que seguir estos sencillos pasos:

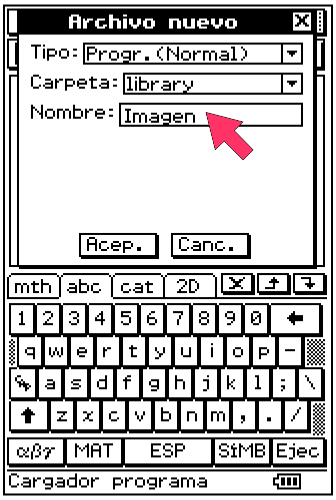
1.- Encienda su Classpad y acceda a la aplicación **Programa** del menú principal.



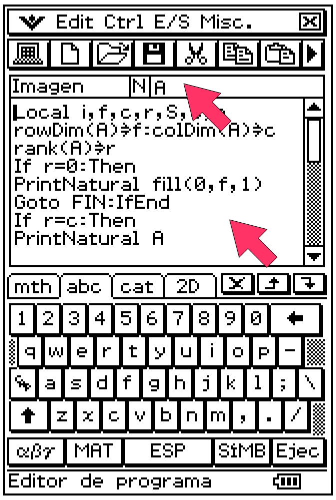
2.- Una vez allí seleccione **Programa Nuevo**



3.- Introduzca el nombre del programa en el campo correspondiente y asegúrese de que tiene seleccionado el tipo **Prog. (Normal)** y la carpeta **library**



4.- Una vez en el editor de programas introduzca en el espacio superior derecho el nombre de la variable correspondiente. En el ejemplo: Imagen(A) → A



Introduzca después el código del programa que encontrará en el apéndice de este tutorial. Asegúrese de respetar los espacios y las mayúsculas del código, así como los retornos de carro y el orden de las líneas.

Apéndice

Imagen(A)

```
Local i,f,c,r,S,c,n
rowDim(A)⇒f:colDim(A)⇒c
rank(A)⇒r:If r=0:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd:If r=c
Then:PrintNatural A
Goto FIN:IfEnd
subMat(A,1,1,f,1)⇒S
rank(S)⇒s:2⇒n:While s<1
subMat(A,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n:WhileEnd
While s<r
subMat(S,1,1,f,s)⇒S
augment(S,subMat(A,1,n,f,n))⇒S
EndIf:n+1⇒n:rank(S)⇒s
WhileEnd:PrintNatural S
Lbl FIN
```

Completa(A)

```
Local i,f,c,r,S,s,n
rowDim(A)⇒f
augment(A,ident(f))⇒A
colDim(A)⇒c:rank(A)⇒r
subMat(A,1,1,f,1)⇒S
rank(S)⇒s:2⇒n:While s<1
subMat(A,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n
WhileEnd:While s<r
subMat(S,1,1,f,s)⇒S
augment(S,subMat(A,1,n,f,n))⇒S
EndIf:n+1⇒n:rank(S)⇒s
WhileEnd:PrintNatural S
Lbl FIN
```

Nucleo(A)

```
Local v,f,c,r,s,B,C,D
Local i,j,k:rowDim(A)⇒f
colDim(A)⇒c:rank(A)⇒r
If r=0:Then
PrintNatural ident(f)
Goto FIN:IfEnd
If r=c:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd
fill(1,1,c)⇒v:1⇒i:0⇒k
While rank(subMat(A,1,i,f,i))=0
0⇒v[1,i]:If k=0:Then
subMat(A,1,i,f,i)⇒C:1⇒k
Else
augment(C,subMat(A,1,i,f,i))⇒C
IfEnd:i+1⇒i:WhileEnd
subMat(A,1,i,f,i)⇒B
rank(B)⇒s:i+1⇒i
While i<(c+1)
augment(B,subMat(A,1,i,f,i))⇒B
If rank(B)=s:Then:
0⇒v[1,i]
subMat(B,1,1,f,s)⇒B
If k=0:Then
subMat(A,1,i,f,i)⇒C
1⇒k:Else
augment(C,subMat(A,1,i,f,i))⇒C
IfEnd:IfEnd
rank(B)⇒s:i+1⇒i:WhileEnd
1⇒i:While i<rowDim(B)+1
If rank(subMat(B,i,1,i,s))=0
Then:If i=1:Then
subMat(B,2,1,rowDim(B),s)⇒B
subMat(C,2,1,rowDim(C),c-s)⇒C
ElseIf i=rowDim(B):Then
subMat(B,1,1,i-1,s)⇒B
subMat(C,1,1,i-1,c-s)⇒C
Else
trn(augment(trn(subMat(B,1,1,i-1,s)),trn(subMat(B,i+1,1,rowDim(B),s))))⇒B
trn(augment(trn(subMat(C,1,1,i-1,c-s)),trn(subMat(C,i+1,1,rowDim(C),c-s))))⇒C
IfEnd:Else:i+1⇒i:IfEnd
WhileEnd:B^(-1)⇒B:B×C⇒C
ident(c-s)×-1⇒B:1⇒k:1⇒j
If v[1,1]=1:Then
subMat(C,1,1,1,c-s)⇒D
k+1⇒k:Else
subMat(B,1,1,1,c-s)⇒D
j+1⇒j:IfEnd:2⇒i:While i<c+1
If v[1,i]=1:Then
trn(augment(trn(D),trn(subMat(C,k,1,k,c-s))))⇒D
k+1⇒k:Else
trn(augment(trn(D),trn(subMat(B,j,1,j,c-s))))⇒D
j+1⇒j:IfEnd:i+1⇒i:WhileEnd
PrintNatural D:Lbl FIN
```

* Las líneas que aparecen en **negrita** NO contienen ningún retorno de carro.

Intersec(A,Z)

```
Local v,f,c,r,s,B,C,D
Local E,S,i,j,k,a,z
If rowDim(A)=rowDim(Z)
Then:colDim(A)⇒a
colDim(Z)⇒z
augment(A,Z)⇒A:Else
Message "Las dimensiones no coinciden","Error!"
Goto FIN:IfEnd
rowDim(A)⇒f:colDim(A)⇒c
rank(A)⇒r:If r=c:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd
fill(1,1,c)⇒v:1⇒i:0⇒k
While rank(subMat(A,1,i,f,i))=0
0⇒v[1,i]:If k=0:Then
subMat(A,1,i,f,i)⇒C:1⇒k
Else
augment(C,subMat(A,1,i,f,i))⇒C
IfEnd:i+1⇒i:WhileEnd
subMat(A,1,i,f,i)⇒B
rank(B)⇒s:i+1⇒i
While i<(c+1)
augment(B,subMat(A,1,i,f,i))⇒B
If rank(B)=s:Then
0⇒v[1,i]
subMat(B,1,1,f,s)⇒B
If k=0:Then
subMat(A,1,i,f,i)⇒C
1⇒k:Else
augment(C,subMat(A,1,i,f,i))⇒C
IfEnd:IfEnd:rank(B)⇒s
i+1⇒i:WhileEnd:1⇒i
While i<rowDim(B)+1
If rank(subMat(B,i,1,i,s))=0
Then:If i=1:Then
subMat(B,2,1,rowDim(B),s)⇒B
subMat(C,2,1,rowDim(C),c-s)⇒C
ElseIf i=rowDim(B):Then
subMat(B,1,1,i-1,s)⇒B
subMat(C,1,1,i-1,c-s)⇒C
Else
trn(augment(trn(subMat(B,1,1,i-1,s)),trn(subMat(B,i+1,1,rowDim(B),s))))⇒B
trn(augment(trn(subMat(C,1,1,i-1,c-s)),trn(subMat(C,i+1,1,rowDim(C),c-s))))⇒C
IfEnd:Else:i+1⇒i:IfEnd
WhileEnd:B^(-1)⇒B:B×C⇒C
ident(c-s)×-1⇒B:1⇒k:1⇒j
If v[1,1]=1:Then
subMat(C,1,1,1,c-s)⇒D
k+1⇒k:Else
subMat(B,1,1,1,c-s)⇒D
j+1⇒j:IfEnd:2⇒i
While i<c+1
If v[1,i]=1:Then
trn(augment(trn(D),trn(subMat(C,k,1,k,c-s))))⇒D
k+1⇒k:Else
trn(augment(trn(D),trn(subMat(B,j,1,j,c-s))))⇒D
j+1⇒j:IfEnd:i+1⇒i:WhileEnd
subMat(D,1,1,a,colDim(D))⇒D
subMat(A,1,1,f,a)⇒A
A×D⇒D:rowDim(D)⇒f
colDim(D)⇒c:rank(D)⇒r
If r=0:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd
If r=c:Then
PrintNatural D
Goto FIN:IfEnd
subMat(D,1,1,f,1)⇒S
rank(S)⇒s:2⇒n:While s<1
subMat(D,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n:WhileEnd
While s<r
augment(subMat(S,1,1,f,s),subMat(D,1,n,f,n))⇒S
EndIf:n+1⇒n:rank(S)⇒s
WhileEnd:PrintNatural S
Lbl FIN
```

- Las líneas que aparecen en **negrita** NO contienen ningún retorno de carro.

Union(A,B)

```
Local i,f,c,r,S,s,n
augment(A,B)⇒A
rowDim(A)⇒f:colDim(A)⇒c
rank(A)⇒r:If r=0:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd:If r=c
Then:PrintNatural A
Goto FIN:IfEnd
subMat(A,1,1,f,1)⇒S
rank(S)⇒s:2⇒n:While s<1
subMat(A,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n
WhileEnd:While s<r
subMat(S,1,1,f,s)⇒S
augment(S,subMat(A,1,n,f,n))⇒S
EndIf:n+1⇒n:rank(S)⇒s
WhileEnd:PrintNatural S
Lbl FIN
```

OrtoGS(A)

```
Local i,f,c,r,S,c,n,T
Local u,v,w:rowDim(A)⇒f
colDim(A)⇒c:rank(A)⇒r
If r=0:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd
subMat(A,1,1,f,1)⇒S
rank(S)⇒s:2⇒n
While s<1
subMat(A,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n
WhileEnd:While s<r
subMat(A,1,n,f,n)⇒u
trn(u)⇒v:1⇒i:While i<s+1
subMat(S,1,i,f,i)⇒w
(u-(det(v×w)/det(trn(w)×w))×w)⇒u
i+1⇒i:WhileEnd
augment(subMat(S,1,1,f,s),u)⇒S
n+1⇒n:rank(S)⇒s:WhileEnd
subMat(S,1,1,f,1)⇒v
(v/√(det(trn(v)×v)))⇒T
2⇒n:While n<s+1
subMat(S,1,n,f,n)⇒v
(v/√(det(trn(v)×v)))⇒v
augment(subMat(T,1,1,f,n-1),v)⇒T
n+1⇒n:WhileEnd
PrintNatural T:Lbl FIN
```

Perpend(A)

```
Local i,f,c,r,S,c,n,T
Local u,v,w:rowDim(A)⇒f
colDim(A)⇒c:rank(A)⇒r
If r=0:Then
PrintNatural ident(f)
Goto FIN:IfEnd
If r=f:Then
PrintNatural fill(0,f,1)
Goto FIN:IfEnd
augment(A,ident(f))⇒A
subMat(A,1,1,f,1)⇒S
rank(S)⇒s:2⇒n:While s<1
subMat(A,1,n,f,n)⇒S
rank(S)⇒s:n+1⇒n
WhileEnd:While s<f
subMat(A,1,n,f,n)⇒u
trn(u)⇒v:1⇒i:While i<s+1
subMat(S,1,i,f,i)⇒w
(u-(det(v×w)/det(trn(w)×w))×w)⇒u
i+1⇒i:WhileEnd
augment(subMat(S,1,1,f,s),u)⇒S
n+1⇒n:rank(S)⇒s:WhileEnd
subMat(S,1,1,f,1)⇒v
(v/√(det(trn(v)×v)))⇒T
2⇒n:While n<s+1
subMat(S,1,n,f,n)⇒v
(v/√(det(trn(v)×v)))⇒v
augment(subMat(T,1,1,f,n-1),v)⇒T
n+1⇒n:WhileEnd
subMat(T,1,r+1,f,f)⇒T
PrintNatural T:Lbl FIN
```