

XCAS pour l' impatient

l'essentiel pour programmer avec XCAS au lycée

Guillaume CONNAN

IREM de Nantes

3 janvier 2010

Sommaire

1 Installation et utilisation

2 Les procédures

3 Tests

4 Boucles

Sommaire

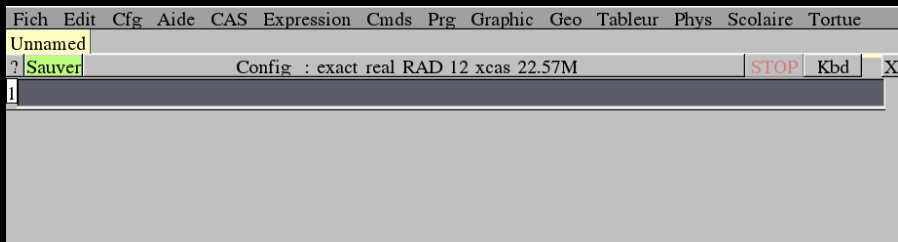
1 Installation et utilisation

2 Les procédures

3 Tests

4 Boucles

Vous récupérez la version pour votre O.S. sur le Site XCAS.
Vous suivez les instructions et voilà...



Il y a beaucoup à dire sur XCAS mais nous nous contenterons d'explorer les outils de programmation.

Nous allons donc ouvrir une fenêtre de programmation en tapant

Alt + P

Une fenêtre de programmation apparaît :

Il y a beaucoup à dire sur XCAS mais nous nous contenterons d'explorer les outils de programmation.

Nous allons donc ouvrir une fenêtre de programmation en tapant

Alt + P

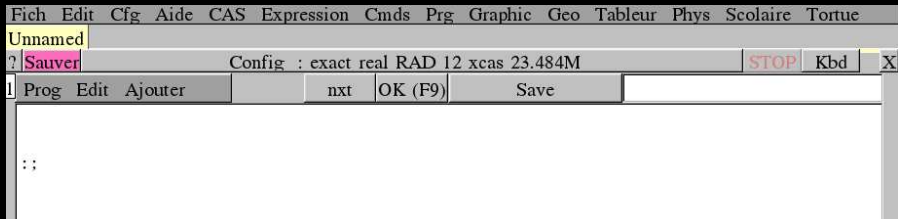
Une fenêtre de programmation apparaît :

Il y a beaucoup à dire sur XCAS mais nous nous contenterons d'explorer les outils de programmation.

Nous allons donc ouvrir une fenêtre de programmation en tapant

Alt + P

Une fenêtre de programmation apparaît :



Il ne reste plus qu'à programmer...

Sommaire

1 Installation et utilisation

2 Les procédures

3 Tests

4 Boucles

XCAS est écrit en C++ et a donc une syntaxe proche de ce langage...mais en bien plus simple. Cependant, comme le C++, l'outil de base est la procédure qui *ressemble* aux fonctions de CAML.

Par exemple, nous voudrions créer une procédure qui prend trois nombres a , b et c en argument et renvoie $b^2 - 4ac$. Appelons-la au hasard...Delta :

XCAS est écrit en C++ et a donc une syntaxe proche de ce langage...mais en bien plus simple. Cependant, comme le C++, l'outil de base est la procédure qui *ressemble* aux fonctions de CAML.

Par exemple, nous voudrions créer une procédure qui prend trois nombres a , b et c en argument et renvoie $b^2 - 4ac$. Appelons-la au hasard...Delta :

Fich Edit Cfg Aide CAS Expression Cnds Prg Graphic Geo Tableur Phys Scolaire Tortue

Unnamed

? Sauver Config : exact real RAD 12 xcas 21.617M STOP Kbd X

1 Prog Edit Ajouter

nxt OK (F9) Save

Delta(a,b,c):={
b^2-4*a*c
}::

Interpreter le programme courant

Par la suite, nous représenterons ce script par :

```
Delta(a,b,c) := {  
  b^2-4*a*c  
};;
```

Il suffit ensuite de cliquer sur OK ou de taper sur 

Par la suite, nous représenterons ce script par :

```
Delta(a,b,c):={  
  b^2-4*a*c  
};;
```

Il suffit ensuite de cliquer sur OK ou de taper sur 

Par la suite, nous représenterons ce script par :

```
Delta(a,b,c):={  
  b^2-4*a*c  
};;
```

Il suffit ensuite de cliquer sur OK ou de taper sur F9

Fich Edit Cfg Aide CAS Expression Cmds Prg Graphic Geo Tableur Phys Scolaire Tortue

Unnamed

1 ? Sauver Config : exact real RAD 12 xcas 21.621M STOP Kbd X

1 Prog Edit Ajouter nxt OK (F9) Save

```
Delta(a,b,c):={  
b^2-4*a*c  
};;
```

// Success compiling Delta

Done M

2

Pour évaluer cette procédure en un certain triplet, nous entrons dans une cellule de calcul :

```
Delta(1,1,1)
```

ce qui apparaît sous la forme :

The screenshot displays a software window with a menu bar (Fich, Edit, Cfg, Aide, CAS, Expression, Cmds, Prg, Graphic, Geo, Tableur, Phys, Scolaire, Tortue) and a title bar (Unnamed). The main area shows a code editor with the following content:

```

1 Prog Edit Ajouter          [nxt] [OK (F9)] [Save]
Delta(a,b,c):={
  b^2-4*a*c
};;

// Success compiling Delta
                               [Done] [M]
2 Delta(1,1,1)
                               [-3] [M]
3
  
```

The code defines a function $\Delta(a,b,c)$ as $b^2 - 4ac$. A comment indicates the compilation was successful. The function is then called with $\Delta(1,1,1)$, resulting in the output -3 .

Sommaire

1 Installation et utilisation

2 Les procédures

3 Tests

4 Boucles

Avec XCAS, on peut programmer en français ou en anglais.

```
val_abs(x):={  
si x>0 alors x sinon -x fsi;  
}::
```

ou

```
val_abs(x) := {  
  if (x > 0) then {x} else {-x}  
};;
```


ou même

```
val_abs(x) := {  
  ifte(x > 0, x, -x)  
};;
```

Sommaire

1 Installation et utilisation

2 Les procédures

3 Tests

4 Boucles

Le bilinguisme est ici aussi de rigueur. Calculons par exemple la somme des premiers entiers :

```
Somme(n) := {  
  local S, k;  
  S := 0;  
  pour k de 1 jusque n faire  
    S := S + k;  
fpour;  
return(S);  
};
```

ce qui apparaît sous la forme :

```

Fich Edit Cfg Aide CAS Expression Cmds Prg Graphic Geo Tableur Phys Scolaire Tortue
Unnamed
? $sauve Config : exact real RAD 12 xcas 22.094M STOP Kbd X
1 Prog Edit Ajouter      nxt OK (F9)      Save
Somme(n) := {
local S,k;
S:=0;
pour k de 1 jusque n faire
    S:=S+k;
fpour;
return(S);
}::

// Parsing Somme
// Success compiling Somme
Done
2 Somme(10)
55
3

```

ou

```
Somme (n) := {  
  local S, k;  
  S := 0;  
  for (k := 1; k <= n; k := k + 1) {  
    S := S + k  
  }  
  return (S);  
}::;
```

OU

```
Somme (n) := {  
  local S, k;  
  S := 0;  
  k := 1;  
  tantque k <= n faire  
    S := S + k;  
    k := k + 1;  
ftantque;  
return(S);  
};;
```

OU

```
Somme (n) := {  
  local S, k;  
  S := 0;  
  k := 1;  
  while (k <= n) {  
    S := S + k;  
    k := k + 1;  
  };  
  return (S);  
}::;
```