

## Stage algorithmique 1

### TI-Nspire

## Pour débiter (fiche 1/3)

### La boucle Pour ....

**Le problème :** le but de cet exercice est de conjecturer une formule donnant la somme des entiers de 1 à  $n$  en fonction de  $n$ .

### 1. Un premier essai à la main

L'idée est de faire découvrir aux élèves la règle de récurrence qui permet de définir la boucle qui sera répétée  $n$  fois :  $S_n = S_{n-1} + n$ .

### 2. La programmation

Algorithme	Programme
$n$ est un argument du programme  $i$ et $s$ sont déclarées comme variables locales <b>Initialiser</b> une variable $s$ à zéro <b>Pour</b> $i$ allant de 1 à $n$ de 1 en 1 Remplacer $s$ par $i + s$ <b>Fin</b> de la boucle Pour <b>Afficher</b> $s$	Define <b>boucle1</b> ( $n$ )= Prgm Local $i,s$ $s:=0$ For $i,1,n,1$ $s:=i+s$ EndFor Disp $s$ EndPrgm

### 3. Comment saisir le programme dans la calculatrice ?

Confection du programme : dans une page **Calculs**, choisir (menu) 9 : Fonctions & programmes / 1 : Éditeur de programmes / 1 : Nouveau. Écrire le nom du programme (**boucle1**) et valider par **OK**.

Les instructions Local et Disp peuvent être tapées directement ou être obtenues à partir du catalogue : 2nd 2nd. Pour certaines d'entre elles, on peut aussi solliciter le menu (voir ci-dessous).

#### Détail de la construction du programme :

Instructions saisies au clavier	Écran	Commentaires
	Define <b>boucle1</b> ( $n$ )=	$n$ est un argument du programme : il suffira de remplacer $n$ par le nombre souhaité en demandant l'application du programme
(menu) 3 : Définir les variables / 1	Prgm Local $i,s$	$i$ et $s$ sont des variables locales : les affectations de leurs valeurs ne sont valables que dans ce programme.
	Prgm Local $i,s$ $s:=0$	On initialise la variable $s$ à 0. $s:=0$ équivaut à $0 \rightarrow s$ , soit 0 (ctrl) (sto+var) $s$ .

Suite du tableau page suivante

Instructions saisies au clavier	Écran	Commentaires
: Contrôle /	Prgm Local $i,s$ $s:=0$ For $i,1,n,1$	On sélectionne l'instruction <b>For...</b> <b>EndFor</b> (boucle).
	Prgm Local $i,s$ $s:=0$ For $i,1,n,1$ $s:=i+s$	On ajoute la valeur de $i$ au contenu de $s$ que l'on stocke dans $s$ . $s:=i+s$ 0 équivaut à $i+s \rightarrow s$ .
: E/S	Prgm Local $i,s$ $s:=0$ For $i,1,n,1$ $s:=i+s$ EndFor Disp $s$ EndPrgm	On descend sous l'instruction de fin de boucle ( <b>EndFor</b> ), pour demander l'affichage du contenu de la variable $s$ ( <b>Disp</b> ).

Quand le programme est écrit, ne pas oublier de l'enregistrer :

: Vérifier la syntaxe et enregistrer / . Raccourci utilisable : .

#### 4. Comment exécuter le programme précédent ?

Passer dans la partie gauche de l'écran : ; si on désire exécuter le programme pour la valeur 4 de  $n$ , écrire boucle1(4) et taper . Le programme s'exécute et affiche la valeur de la somme

$$S = 1 + 2 + 3 + 4 = 10.$$

Pour obtenir un autre calcul (pour  $n = 7$ , par exemple), retourner, avec la flèche haute sur *boucle1(4)*, taper , remplacer 4 par 7 puis taper .

<i>boucle1(4)</i>
10
Terminé
<i>boucle1(7)</i>
28
Terminé

#### 5. Comment corriger un programme qui comporte une erreur de syntaxe ?

	On a vérifié la syntaxe d'un programme qui comporte une erreur ; le logiciel détecte l'erreur.
$s:=0$ For $i, n, 1$ $s:=i+s$	Le curseur s'est positionné sur l'erreur commise. Ici, il manque le 1 du début de la boucle Pour.

On insère le chiffre 1 manquant.

On vérifie et enregistre le programme modifié :  $\text{menu}$   $\text{2}$  : Vérifier la syntaxe et enregistrer /  $\text{1}$ .

## 6. Comment modifier un programme existant ?

On souhaite par exemple ici afficher, après la valeur de  $s$ , la valeur de l'expression  $\frac{n(n+1)}{2}$  afin de la comparer à  $S$ .

Se placer dans la partie droite de l'écran, où figure le programme à modifier, au besoin avec  $\text{ctrl}$   $\text{tab}$ .

Placer le curseur juste après  $\text{Disp } s$  et appuyer sur  $\text{enter}$  pour insérer une ligne.

```
Disp s
[ ]
EndPrgm
```

Taper  $\text{Disp } n*(n+1)/2$ .

( $\text{Disp}$  peut aussi être obtenu par :  $\text{menu}$   $\text{6}$  : E/S  $\text{1}$ ).

On obtient l'insertion de ligne ci-contre.

```
Disp s
Disp  $\frac{n \cdot (n+1)}{2}$ 
EndPrgm
```

On vérifie et enregistre le programme modifié.