TP n°6 Somme d'entiers

I/ Calcul d'une somme

- 1°) Calculer la somme suivante (par exemple avec une calculatrice) : 1+2+3+4+5+6+7+8+9+10=.....
 Le but de ce TP est d'explorer un algorithme permettant de calculer des sommes de ce genre, allant jusqu'à 1+2+3+...+9999+10000!
- 2°) Démarrer le tableur *Open Office Calc*, et remplir la colonne A avec les entiers allant de 1 à 10 (1 dans A1, etc. et 10 dans A10).
- 3°) Calculer dans la cellule B1, la somme des éléments de la colonne A (on pourra utiliser le bouton $|\Sigma|$).

II/ Tableur et JavaScript

On va utiliser une boucle pour additionner les entiers successifs. Pour cela, on crée une variable *somme* qu'on initialise à 0, et à laquelle on va successivement additionner les valeurs de l'indice de la boucle :

Algorithme 0.1: ADDITION DES ENTIERS()
variables entières somme, indice
somme ← 0
pour indice ← 1 jusqu'à 8
faire somme ← somme + indice

1°) Dans l'onglet "Outils", sélectionner "Macros" puis "Gérer les macros", et enfin choisir "JavaScript" :

HBC	Vérification orthographique F7 Langue Audit AutoCorrection		ؤ 2	3 2↓ 7↓ 0 6 \$% .∷∋ 09) / M (‡ (‡	⊘ 💼 🖾 ⊘ □ • ७ • A	\ C
7	R <u>e</u> cherche de valeur cible						
	<u>S</u> cénarios	F		G	н	T	
	Protection	• 🗖				-	
	Contenu des cellules	•					
â	<u>G</u> allery						
111	Media Pla <u>v</u> er						
	<u>M</u> acros	•	Ēr	nregistrer une ma	cro		
	Gestionnaire des <u>e</u> xtensions		E>	kéc <u>u</u> ter la macro.	. <u>L</u>		
	Paramétrage du filtre XML		Ge	érer les macr <u>o</u> s	StarOffice <u>B</u> asic		
	<u>P</u> ersonnaliser Opt <u>i</u> ons			gnature numériqu	JavaScript		
					Bea <u>n</u> Shell	- 1	

2°) Cliquer sur "créer" pour créer un dossier de macros JavaScript, qui s'appellera Algorithmique :



3°) Cliquer à nouveau sur "Créer" pour créer un fichier JavaScript, qu'on appellera Somme :



- 4°) Cliquer enfin sur "Éditer" pour ouvrir la fenêtre JavaScript.
- 5°) Enlever tout le JavaScript qui est dans la fenêtre en question, puis le remplacer par le script suivant :

var somme=0;
<pre>for(var indice=0;indice<=8;indice++){</pre>
<pre>somme+=indice;</pre>
}

6°) Pour tester l'algorithme, cliquer sur "File>Run". Ceci permet d'afficher la valeur des variables *indice* et *somme* à des instants successifs. En bas à droite, dans l'onglet "Watch", cliquer sur la première case, entrer *indice*, cliquer sur la case "value" correspondant à *indice*, ce qui crée une case vide en-dessous. Enfin écrire dans cette case vide le nom de la variable *somme* :

📓 Rhino JavaScript Debugger 📃 🗖 🗙												
<u>F</u> ile	<u>E</u> dit <u>D</u>	ebug <u>V</u>	<u>V</u> indow									
	Break		Go	Step I	nto	•	Step Ov	er	Step	Out		
📄 vnd.sun.star.expand:\${\$BRAND_BASE_DIR/program/bootstraprc::UserIn 🖬 🖆 🔟												
<pre>1 ⇔var somme=0; 2 for(var indice=0;indice<=8;indice++){ 3 somme+=indice; 4 }</pre>												
Co	ontext:	"js/ess	sai2.js/ucb/	", line 1	-		Expre	ession		V	'alue	
	Name proto		Va [object Obj	ue ect]	e ct]		somme					
	this Locals						Watch	Eval	uate			
Thread: Thread[Thread-220,6,main]												

7°) Pour avancer d'un pas, cliquer sur "Step Into". Remplir alors le tableau suivant :

indice	somme

III/ Question non notée, seulement pour les rapides

Essayer de trouver une formule qui permet d'obtenir la somme des n premiers entiers, soit 1 + 2 + 3 + ... + n.

- 1°) Avec le tableur, on peut regarder les doubles des sommes (dans une nouvelle colonne).
- 2°) Avec *GeoGebra*, on peut placer des points dans le repère, ayant pour coordonnées celles qui figurent dans le tableau ci-dessus, puis utiliser l'outil *Régression polynomiale* en sélectionnant ces points.
- 3°) Avec Xcas, on peut entrer simplify(sum(k,k,1,n))...