**ASTUCES & UTILITAIRE:**-Choppe un tableau à partir de c :  
Range ("B3:" & Range("B3").End(xlDown).End(xlToRight)**.Address**)  
  
-Forcer choix Inputbox : 0 formule**, 1 nombre**, 2 string, 8 plage cellule  
- Effacement **DEPUIS LIGNE** (sinon UsedRange) :  
Range("A" & n).Resize(**Rows.Count - n**, Columns.Count).Clear  
  
- Dernière cell active, écris N dans celle d'en dessous (sinon utiliser j = j+1 si condition remplie):  
Set rg = Cells(**Rows.Count**, 1).End(**xlUp**)  
rg.Offset(1, 0).Value = N  
  
- DateAdd pour ajouter un nombre de jours à une date : DateAdd(interval, number,date)  
- Remplacer dans une plage : c.Resize(1,n).Replace **what :=**"QUOI" **Replacement :=** "PAR CA"  
- Remplacer dans une plage : c.Value = **Replace**(c.value, "QUOI","PAR CA")  
- UCase() et LCase() : MAJUSCULE ou minuscule

- Reporter sur la feuille calcul le nom des indices / !\ UTILE  
wsC.Range("A2").Resize(nbre, 1).Value = WorksheetFunction.Transpose(**wsD.Range("B1").Resize(1, nbre)**)  
 **FORMULES :**  
Cells(i,j).FormulaR1C1 = "sum(" & **rg.address(ReferenceStyle :=xlR1C1) & ")** " 🡪 =somme(L2C1 :L6C2)  
Cells(i,j).FormulaArray = "Sum(R1C1 :R3C3) "  
Cells(i,j).value = Application.WorksheetFunction**.CountIf(Rg, ">0")**  
Cells(i,j).value = Application.WorksheetFunction**.SumIf(RgR, "<0", RgR)**Cells(i,j). Formula**Array** = "=sum(if(RC[-1] = produit, quantité\*prix, 0))"  
  
**EXTRAIT D’EXAMENS :**-rg récupère la cellule de ws2 contenant la donnée à chercher, dans la matrice en ws2. La valeur de départ se trouve dans ws1.  
rgC est la matrice dans laquelle on cherche.  
Set **rg** = ws2.Rows(1).Find(ws1.Cells(4, i + 1))  
ws1.cValue = WorksheetFunction.VLookup(ws1.Cells(j + 4, 1), rgC, rgIS.Column)  
  
NB : 'La ligne suivante devrait fonctionner, ma la formule ne considère pas que la date est une date  
 wsS.Cells(4 + j, 1 + i).FormulaLocal = "=rechercheV(" & CStr(wsS.Cells(4 + j, 1)) & ";" & rgC.Address(external:=True) & ";" & col & ";0)"

'Calculez le montant des frais de gestion : utilisez la fonction **VLOOKUP**Cells(i + 2, 7).**FormulaR1C1** = "=VLOOKUP(**RC[-3],Frais\_Gestion,RC[-4]+1)\*RC[-1]**"

**'tri décroissant** de rg par rapport au rendement moyen (avec SORT)  
rg**.Sort Key1:**=wsC.Range("B1"), **Order1:=**xlDescending, **Header:=**xlYes

'Calculez dans la cellule C le montant des transactions effectuées par le client choisi  
C.Value = WorksheetFunction.**SumIf**(Range("Clients"), Client, Range("Montants"))

'Calculer les statistiques de l'indice i  
wsSt.Cells(3, i + 1).Value = WorksheetFunction.**CountIf**(RgR, ">0")  
wsSt.Cells(5, i + 1).Value = WorksheetFunction.**SumIf**(RgR, ">0") / wsSt.Cells(3, i + 1).Value

'Reporter la somme correspondant à la valeur des titres liquidés (valeur =0 après ajustement ; ne pas utiliser de formule, fonction Excel SumIf)  
C.Value = WorksheetFunction.SumIf(**Range(**"ValApRealloc"**)**, "=0", **Range(**"ValT"**)**)

* La fonction **SolverOK** permet de définir un modèle élémentaire du solveur. Elle a 4 paramètres :
  + **SetCell** fait référence à une cellule de la feuille active. Cette cellule contient nécessairement une formule qui dépend des cellules variables.
  + **MaxMinVal** correspond aux options Max, Min et Valeur de la boite de dialogue « Paramètres du solveur ». Ce paramètre prend les valeurs 1 (Max), 2 (Min) ou 3 (Valeur).
  + Quand **MaxMinVal** prend la valeur 3, **ValueOf** permet de spécifier la valeur à laquelle doit être égale la cellule cible.
  + **ByChange** indique quelles sont les cellules variables
* La fonction **SolverAdd** permet d’ajouter une contrainte au modèle. Elle a 3 paramètres :
  + **CellRef** fait référence à la cellule qui se trouve du côté gauche de la contrainte.
  + **Relation** indique la nature de la relation qui existe entre le côté gauche et le côté droit de la contrainte. Ce paramètre prend les valeurs 1 (<=), 2 (=), 3 (>=), 4 (**CellRef** doit être un nombre entier) ou 5 (**CellRef** doit être égal à 0 ou 1)
  + **FormulaText** fait référence à la valeur ou à la cellule qui se trouve du côté droit de la contrainte.
* La fonction **SolverSolve** lance la résolution du problème. Elle correspond au bouton **Résoudre** de la boite de dialogue « **paramètre du solveur** ».. Elle a 2 paramètres.
  + **UserFinish.** Lorsque ce paramètre prend la valeur **True**, le solveur renvoie le résultat sans afficher la boite de dialogue **Résultat du Solveur** (cf. diapo 14).
  + **ShowRef** (cf. aide)
* Modifiez le code de Macro2 en attribuant la valeur **True** au paramètre **UserFinish**. Vérifiez que la boite de dialogue **Résultat du Solveur** ne s’ouvre plus.
* FORMULES

Revenons à l’exemple précédent. Affectons la plage B2:B6 à la variable rg :

Set rg= Range(“B2:B6“)

Range("B7").Formula = "=sum(" & rg.Address & " )"

Ce code reporte dans la cellule B7 la formule

= somme($B$2:$B$6)

Range("B7").formula=“=sum(“ & rg.address(false,false) & “)

Ce code reporte dans la cellule B7 la formule

= somme(B2:B6)