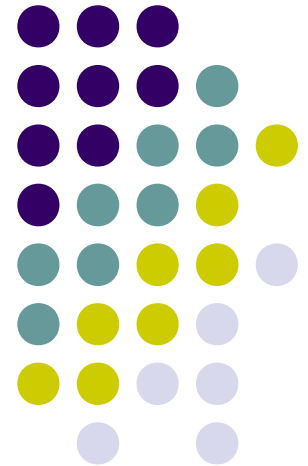


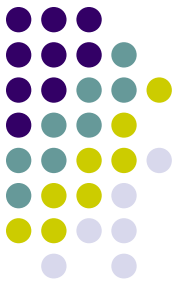
Outils informatiques 2

Cours n° 5 : Structures de contrôle répétitives (suite)

Claire Hanen

Juliette Arnal





Variante 2 de la boucle While

Syntaxe :

Do Until expr

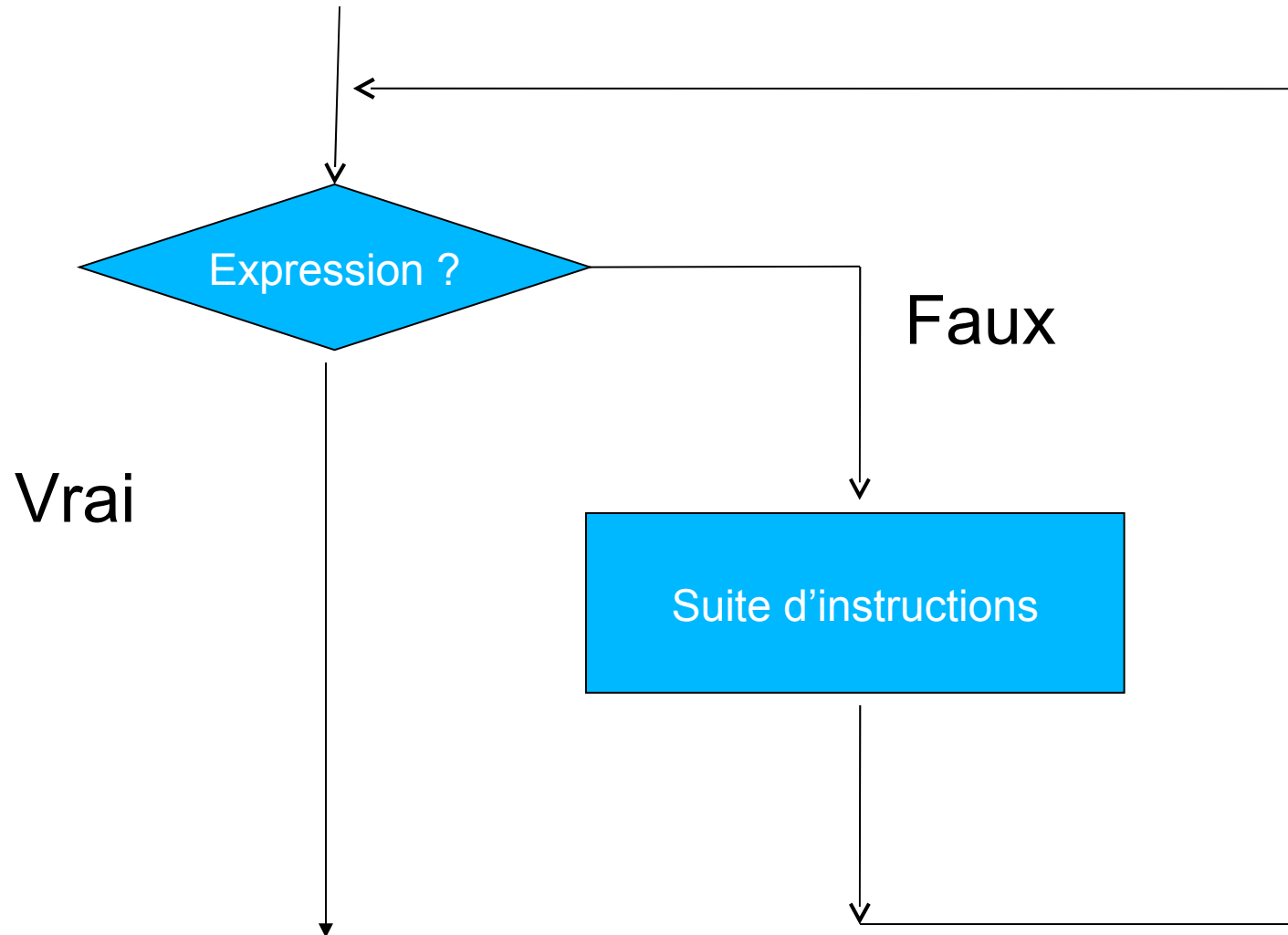
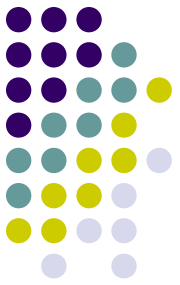
Séquence d'instructions

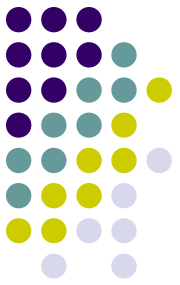
Loop

Ici, le principe est le même, mais on utilise UNTIL (Jusqu'a ce que). La condition exprimée est donc celle de sortie de boucle

Instruction répétitive

Do Until...Loop



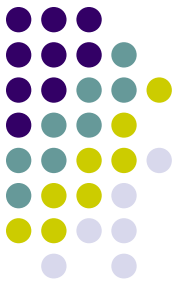


Somme des carrés

- Sub somcarre()
- 'calcule la somme des carrés des entiers de 1 au nombre donné par l'utilisateur
- Dim max As Integer
- Dim n As Integer
- Dim somme As Integer
- max = InputBox("entrez un entier")
- somme = 0
- n = 1
- Do until n > max
- somme = somme + n * n
- n = n + 1
- Loop
- MsgBox ("la somme des carrés des entiers de 1 à " & max & " est : " & somme)
- End Sub

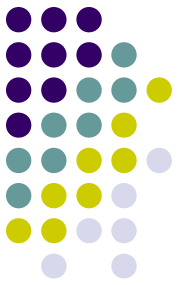
Condition de sortie

Rappel: Do While n <= max



Exemple Do Until...Loop

```
Sub MotPasse()  
    ' Déclaration des variables  
    Dim Reponse As String ' Contient l'essai de mot de passe  
    Dim NombreEssai As Integer ' Compte les essais  
  
    ' Première demande :  
    Reponse = InputBox("Entrez le mot de passe")  
    ' De tout façon, on doit faire au moins 1 essai  
    NombreEssai = 1  
  
    ' Faire la boucle JUSQU'A CE QUE L'utilisateur entre youpi  
    Do Until Reponse = "youpi"  
        Reponse = InputBox("Mauvaise réponse. Essayez encore")  
        ' On augmente le nombre d'essais :  
        NombreEssai = NombreEssai + 1  
    Loop ' Fin de la boucle  
  
    ' Message de bienvenue final. Si c'est trouvé en un essai,  
    ' écrire au singulier, et écrire au pluriel  
    ' si c'est en plusieurs essais :  
    If NombreEssai = 1 Then  
        MsgBox (« Mot de passe trouvé au premier Essai »)  
    Else  
        MsgBox (« Mot de passe trouvé en " & NombreEssai & " Essais » )  
    End If  
End Sub
```



Variante 3 de la boucle While

Syntaxe :

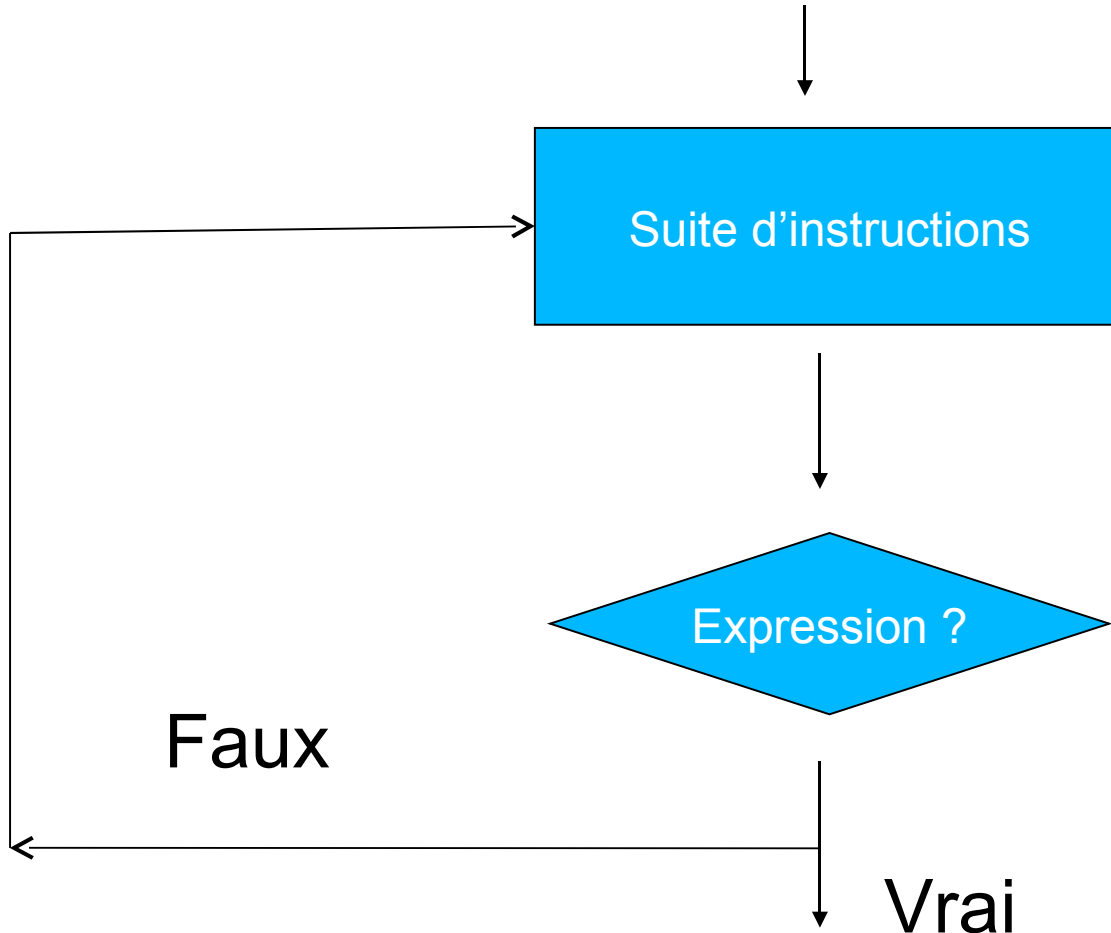
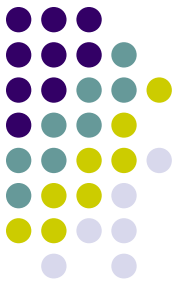
Do

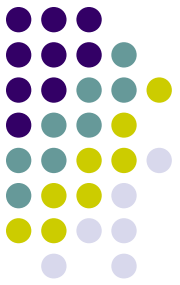
Séquence d'instructions

Loop Until expr

Instruction répétitive

Do ... Loop Until





Jeu de l'addition répété

- nbadd = 0
- scoreglobal = 0
- Do
 - $x = \text{Int}(\text{Rnd}() * 100) + 1$
 - $y = \text{Int}(\text{Rnd}() * 100) + 1$
 - $\text{nbadd} = \text{nbadd} + 1$
 - $\text{nb} = 0$
 - $\text{score} = 5$
 - $z = \text{InputBox}(\text{"combien font " \& x \& " et " \& y \& " ?"})$
- Do until (nb= 5) Or (z = x + y)
 - $\text{score} = \text{score} - 1$
 - $z = \text{InputBox}(\text{"Erreur, encore un essai: combien font " \& x \& " et " \& y \& " ?"})$
 - $\text{nb} = \text{nb} + 1$
- Loop
 - If (z = x + y) Then
 - $\text{MsgBox}(\text{"Bravo!"})$
 - Else
 - $\text{MsgBox}(\text{"La bonne réponse était: " \& x + y})$
 - End If
- $\text{scoreglobal} = \text{scoreglobal} + \text{score}$
- $\text{continue} = \text{InputBox}(\text{"Voulez-vous une nouvelle addition (oui/non) ?"})$
- Loop Until (continue = "non")
- $\text{note} = (20 * \text{scoreglobal}) / (5 * \text{nbadd})$
- $\text{MsgBox}(\text{" le jeu est fini, votre score est de : " \& scoreglobal \& " sur " \& nbadd \& "additions" \& " et votre note est : " \& note \& "/20"})$

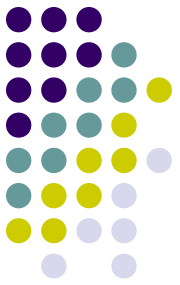
La boucle interne s'arrête dès que l'utilisateur donne la bonne réponse ou qu'il a fait 5 essais.

La boucle externe s'arrête lorsque l'utilisateur répond « non »

S'il répond autre chose que non, par ex no, une nouvelle addition est proposée.

Structure répétitive

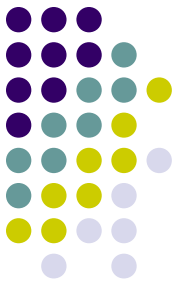
Boucle For ...Next



- L'instruction For Next permet de répéter des actions un nombre de fois prédéfini.
- Vous devez spécifier une valeur de début et une valeur de fin. La variable compteur va ensuite être incrémentée ou décrétementée à chaque itération.
- Pour utiliser cette structure de contrôle, on doit avoir une variable de type numérique (Byte, integer) qui sert à compter le nombre d'itérations de la boucle, et l'on doit **connaître ce nombre à l'avance**.

Structure répétitive

Boucle For ...Next



Syntaxe simplifiée:

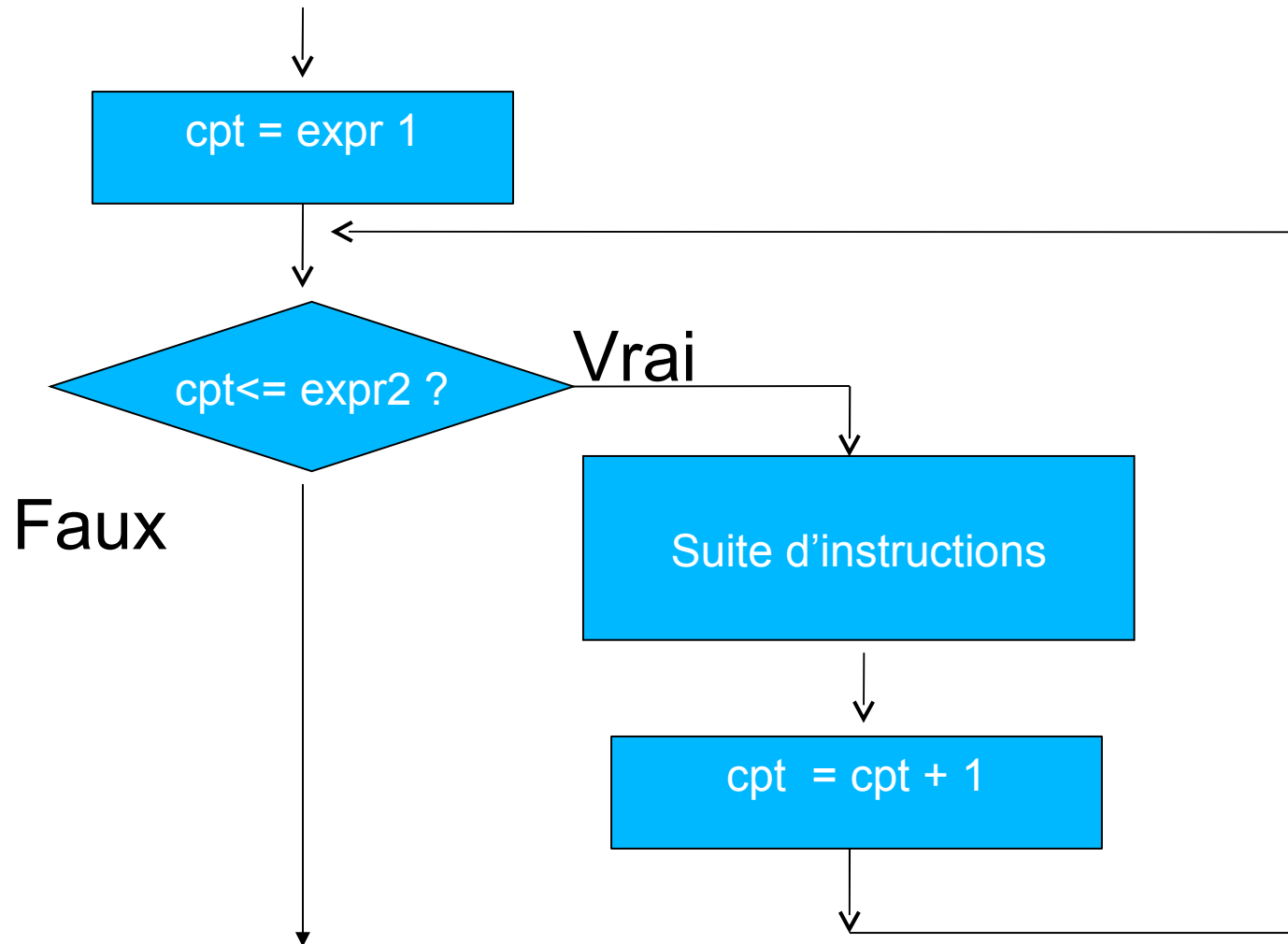
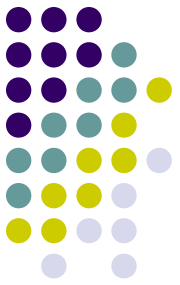
```
For cpt=expr1 To expr2  
Séquence d'instructions  
Next
```

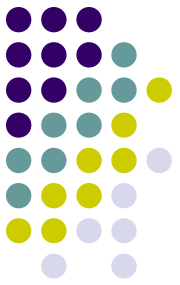
cpt est la variable compteur

expr1 et expr2 sont deux expressions de même type que cpt (ou convertissable)

Instruction répétitive

Boucle For ...Next

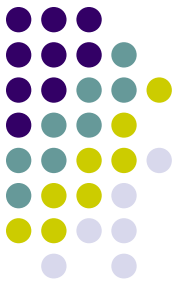




Exemple For ...Next

```
Sub moyenne()  
    Dim x, somme as double  
    Dim i as integer  
    somme = 0  
    For i = 0 to 19  
        x = inputBox("Entrer la note :" )  
        somme = somme + x  
    next  
    MsgBox("la moyenne est : " & somme/20)  
End sub
```

i est la variable
compteur



Somme des carrés

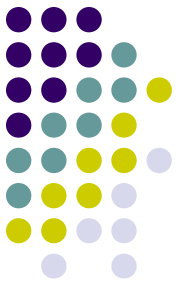
- Sub somcarre()
- 'calcule la somme des carrés des entiers de 1 au nombre donné par l'utilisateur
- Dim max As Integer
- Dim n As Integer
- Dim somme As Integer
- max = InputBox("entrez un entier")
- somme = 0
- For n=1 to max
- $\text{somme} = \text{somme} + n * n$
- Next
- MsgBox ("la somme des carrés des entiers de 1 à " & max & " est : " & somme)
- End Sub

La variable n varie de 1 à max

⇒ On peut utiliser la boucle for

On n'a pas besoin d'incrémenter (ajouter 1 à) la variable n à l'aide d'une instruction

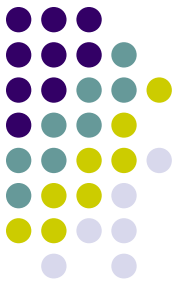
Boucle For ...Next – points d'attention!



- Il ne faut pas modifier la variable compteur dans l'itération:
 - For i=1 to n
 - instructions
 - i=i-1
 - Next ' **produit une boucle infinie!!!**
- Si au cours de la séquence d'instructions (par exemple dans l'une des alternatives d'un If ou d'un Select) on insère une instruction **Exit For**, cela revient à interrompre la boucle et à passer à l'instruction suivante.

Structure répétitive

Boucle For ...Next



Syntaxe générale :

```
For cpt= expr1 To expr2 [Step pas]
```

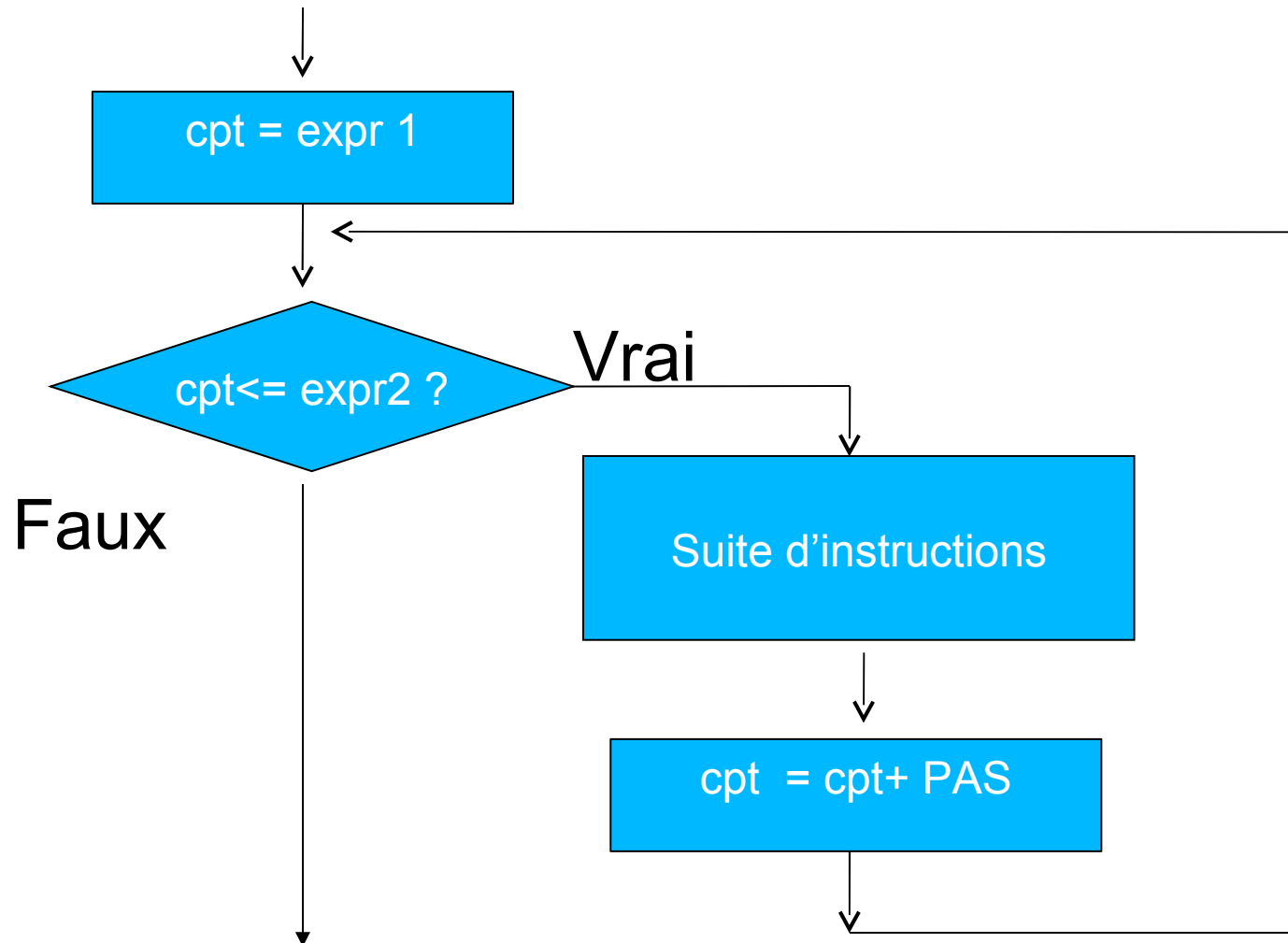
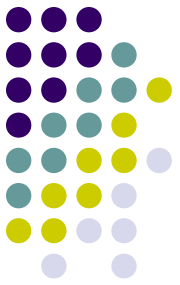
```
Séquence d'instructions
```

```
Next
```

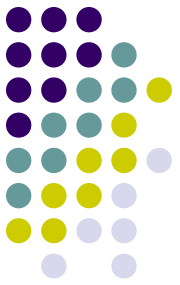
Pas est une expression de même type que cpt

Si elle n'est pas indiquée, par défaut elle vaut 1

Organigramme lorsque pas>0

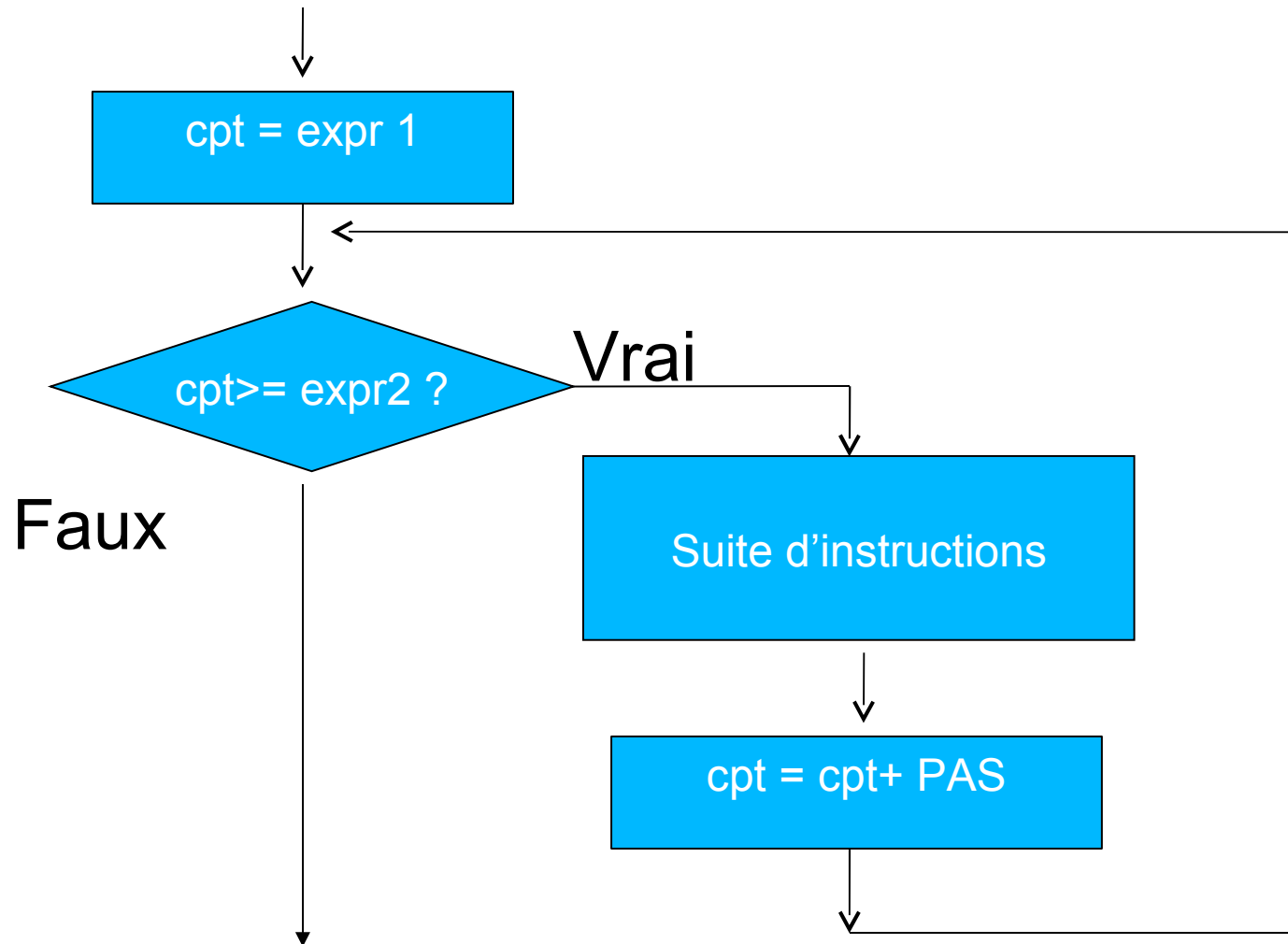
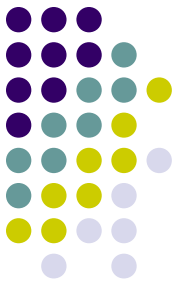


Somme des carrés des impairs

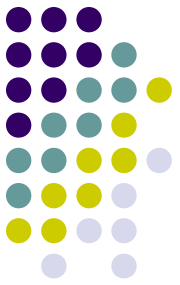


- Sub somcarreimpair()
- 'calcule la somme des carrés des entiers impairs de 1 au nombre donné par l'utilisateur
- Dim max As Integer
- Dim n As Integer
- Dim somme As Integer
- max = InputBox("entrez un entier")
- somme = 0
- For n=1 to max step 2
- somme = somme + n * n
- Next
- MsgBox ("la somme des carrés des impairs de 1 à " & max & " est : " & somme)
- End Sub

Organigramme lorsque pas<0



Somme des carrés des entiers calculée à l'envers



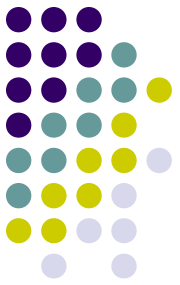
- Sub somcarreenv()
- 'calcule la somme des carrés des entiers de 1 au nombre donné par l'utilisateur
- Dim max As Integer
- Dim n As Integer
- Dim somme As Integer
- max = InputBox("entrez un entier")
- somme = 0
- For n=max to 1 step -1
- *somme = somme + n * n
- Next
- *MsgBox ("la somme des carrés des impairs de 1 à " & max & " est : " & somme)
- End Sub

Exemple: max=5

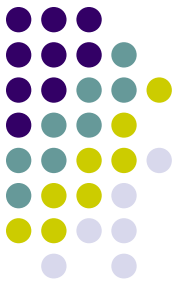
N	5	4	3	2	1	0
somme	0	25	41	50	54	55

A chaque itération *, somme vaut la somme des carrés des entiers de n+1 à max

Exemple sur les chaînes de caractères.



- Rappel: la fonction `mid(mot, pos, lg)` renvoie une chaîne extraite de `mot` à partir de la position `pos` sur `lg` caractères
- Exemple: `mid("Exemple",2,1)` vaut "x "
- Concevoir un programme qui compte le nombre de voyelles d'un mot ou d'une phrase entrée par l'utilisateur et l'affiche.



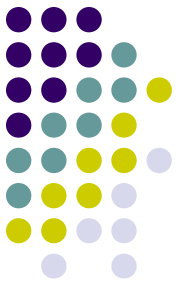
Nombre de voyelles

- Sub nbvoyelle()
- Dim mot As String, lettre As String
- Dim i As Integer, nbv As Integer
- nbv = 0
- mot = InputBox("Entrez un texte")
- For i = 1 To Len(mot)
- lettre = Mid(mot, i, 1) ' lettre est le ième caractère du mot
- Select Case UCase(lettre)
- Case "A", "E", "I", "O", "U"
- nbv = nbv + 1
- End Select
- Next
- MsgBox ("votre texte comporte " & nbv & " voyelles")
- End Sub

Dans le select case, on peut lister différentes valeurs séparées par des virgules (ce qui n'est pas possible avec if)

Chaque fois que le caractère examiné est une voyelle, on incrémente la variable nbv qui sert de compteur

Exemple: calcul du nombre de combinaisons



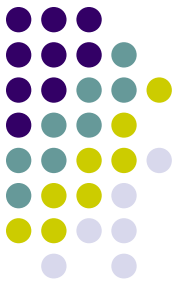
- Sub Cnp()
 - Dim n As Integer
 - Dim p As Integer
 - Dim i As Integer
 - Dim res As Double
 - n = InputBox("entrez un entier")
 - p = InputBox("entrez un entier inférieur à " & n)
 - res = 1
 - For i = 0 To p - 1
 - res = res * (n - i) / (p - i)
 - Next
 - MsgBox ("Le nb de tirages de " & p & " éléments parmi " & n & " est : " & res)
 - End Sub

Calcul du nombre de combinaisons:

$$C(n,p) = n!/(p!(n-p)!)$$

$$C(n,p)=(n/p)*(n-1)/(p-1)*...(n-p+1)/1$$

Calcul et affichage du triangle de Pascal?



1 1

Les coefficients qui apparaissent ligne n sont ceux qui apparaissent dans le développement de $(a+b)^n$

1 2 1

1 3 3 1

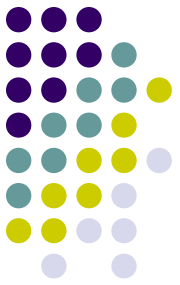
1 4 6 4 1

1 5 10 10 5 1

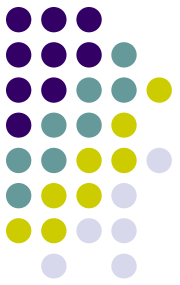
Ligne n colonne p , on trouve le coefficient $C(n,p)$

Il faut donc répéter le calcul de $C(n,p)$ pour différentes valeurs de n et de p

Résoudre un problème : le triangle de Pascal



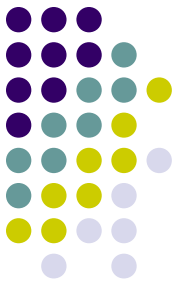
- Décomposer le problème en étapes (en français):
 - Demander à l'utilisateur la taille du triangle qu'il souhaite afficher (max)
 - Pour n allant de 1 à max
 - Calculer la ligne n et l'ajouter au triangle
 - passer à la ligne dans le triangle
 - Afficher le triangle
- Pour calculer la ligne n et l'ajouter au triangle:
 - Pour p allant de 0 à n
 - Calculer le coefficient $C(n,p)$
 - L'ajouter au triangle
- Traduire ensuite les différentes étapes en VBA



Boucles For imbriquées

- Lorsque deux boucles sont imbriquées l'une dans l'autre, on ajoute, suite au Next, le nom de la variable compteur:
- For i=1 to n
 For j=1 to p
 instructions
 Next j
Next i

Résolution triangle de Pascal



- Sub Pascal()
- Dim max As Integer, n As Integer, p As Integer, i As Integer
- Dim res As Double
- Dim triangle As String
- max = InputBox("Entrez le nombre de lignes du triangle")
- triangle = ""
- For n = 1 To max ' calcul de la ligne n
- For p = 0 To n ' calcul de $C(n,p)$ pour le n et le p fixé
- res = 1
- For i = 0 To p - 1
- res = res * (n - i) / (p - i)
- Next i
- triangle = triangle & " " & res ' On l'ajoute au triangle
- Next p
- triangle = triangle & vbCrLf ' On passe à la ligne dans la chaine triangle
- Next n
- MsgBox (triangle)
- End Sub