

TD3 : boucles FOR

ECE3 Lycée Carnot

15 octobre 2010

Boucles FOR

Ce TD va vous rappeler de bons souvenirs de cours de maths, puisque le principe d'une boucle FOR en Pascal est assez proche de celui du symbole \sum en maths, à savoir supprimer des petits points. Une boucle FOR est donc une instruction répétitive permettant d'effectuer plusieurs fois de suite des calculs similaires. Des exemples classiques en maths sont les calculs de sommes, mais aussi les calculs de termes d'une suite définie par une formule de récurrence. L'instruction FOR obéit à la syntaxe suivante :

```
FOR i :=1 to n DO instruction ;
```

Comme en mathématiques quand on manipule une somme, la variable i est muette et vous pouvez donc lui donner n'importe quel autre nom. Elle devra bien sûr, comme toute variable, être déclarée en début de programme, avec un type integer. Toujours comme en maths, votre variable prendra toutes les valeurs entières entre la valeur initiale et la valeur finale stipulées. Les instructions placées à l'intérieur de la boucle peuvent être des calculs faisant intervenir la variable i (quand on calcule par

exemple $\sum_{i=1}^{i=12} i^2$), mais il est fortement déconseillé de modifier la valeur de i à l'intérieur de la boucle

sous peine d'obtenir des bugs étranges (notez que i augmente tout seul à l'intérieur de la boucle). Dernier détail, on a le droit de faire des boucles où la valeur de i diminue au lieu d'augmenter, il suffit de remplacer dans la syntaxe le mot TO par DOWNTO. Petit exemple de programme utilisant une boucle FOR :

```
PROGRAM suite ;
VAR u : real ;
i,n : integer ;
BEGIN
WriteLn('Choisissez la valeur de n') ;
ReadLn(n) ;
u := 1 ;
FOR i := 1 to n do u := u/2 + 1/u ;
WriteLn('u_n=',u) ;
END.
```

Dans le cas où l'on souhaite effectuer plusieurs instructions à chaque étape de la boucle, on encadrera ces instructions par un BEGIN et un END (sinon, seule la première sera effectuée à chaque étape, et les suivantes seulement quand la boucle sera terminée).

Petits exercices

1. Écrire un programme calculant $\sum_{k=1}^{k=n} \frac{1}{k}$, pour une valeur de n choisie par l'utilisateur.
2. Écrire un programme calculant $n!$, pour une valeur de n choisie par l'utilisateur.
3. Écrire un programme calculant, pour un entier n choisi par l'utilisateur, la valeur du terme d'indice n de la suite de Fibonacci (définie, rappelons-le, par $u_0 = u_1 = 1$, puis $u_{n+2} = u_{n+1} + u_n$).
4. Écrire un programme qui calcule et affiche les n premières valeurs de la suite définie par
$$u_n = \sum_{k=0}^{k=n} \frac{1}{k!}.$$