

## TP 5

---

### Exercice 1 :

Programmer l'exercice 1 du TD 4. Rappel de l'énoncé :

*Écrire un algorithme permettant d'afficher tous les nombres parfaits compris entre 1 et n, avec n saisi par l'utilisateur.*

*Rappel : un nombre est dit parfait lorsqu'il vaut la somme de tous ses diviseurs (lui-même exclu).*

*Exemple : 6 est un nombre parfait, car  $6 = 1 + 2 + 3$*

***Présenter cet algorithme sous la forme d'une fonction ainsi qu'un programme l'utilisant***

### Exercice 2 :

Programmer l'exercice 2 du TD 4. Rappel de l'énoncé :

*Écrire une fonction permettant de vérifier qu'une date est valide (déjà fait en TP) sachant que :*

*- on se place exclusivement dans le cadre du calendrier grégorien, à partir de l'année 1583 (les dates plus anciennes seront considérées comme non valides).*

***- aucun "si alors sinon" n'est autorisé dans la fonction***

*- la fonction doit être utilisable par le programme suivant :*

```
programme verification_date
debut
    ...
fin verification_date
```

### Exercice 3 :

Écrire un programme qui demande à l'utilisateur d'entrer une valeur de **n** strictement positive puis affiche le « triangle » suivant (les bords doivent être "droits") :

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
...
...
...
1 2 3 4 5 6 7 ... n
```

### Exercice 4 :

Écrire un algorithme qui lit une valeur n entrée par l'utilisateur, puis affiche un carré de nombres de côté  $2n + 1$ , composé de la manière suivante (exemple donné avec  $n = 4$ ) :

```
8 7 6 5 4 5 6 7 8
7 6 5 4 3 4 5 6 7
6 5 4 3 2 3 4 5 6
5 4 3 2 1 2 3 4 5
4 3 2 1 0 1 2 3 4
5 4 3 2 1 2 3 4 5
6 5 4 3 2 3 4 5 6
7 6 5 4 3 4 5 6 7
8 7 6 5 4 5 6 7 8
```