

Exercices de révision - DUT S2

Exercice 1. (bourses)

Pour avoir droit à une bourse :

- il faut que j'aie moins de 26 ans (à moins que j'ai fait mon service militaire, dans ce cas j'ai droit à 1 an de plus) ;
- il faut que le revenu imposable annuel de mes parents ne dépasse pas 1200€ plus 40€ par enfants, le tout multiplié par 12 ;
- si on est issu d'une famille de 3 enfants ou plus, la bourse est accordée sans autre condition.

Écrire la ou les instructions qui permettent d'afficher "Bourse accordée" ou "Bourse refusée" selon la valeur des paramètres.

Exercice 2. (mi-carré – mi-cube)

Un nombre entier positif est dit *mi-carré mi-cube* s'il peut s'écrire comme la somme d'un carré et d'un cube (entiers positifs non nuls). Ainsi, l'an 2000 est une année *mi-carrée mi-cube* car $2000 = 44^2 + 4^3$ de même que l'an 2 car $2 = 1^2 + 1^3$.

Écrire un algorithme qui affiche les années *mi-carrées mi-cubes* < 2017 .

Exercice 3. (nombres parfaits)

Un nombre parfait est un entier > 1 qui est égal à la somme de ses diviseurs (y compris 1, mais non compris lui-même). Par exemple, 6 est parfait car $6 = 1 + 2 + 3$. Ecrire un algorithme qui énumère tous les nombres parfaits inférieurs à un entier donné.

Exercice 4. (distributeur de billets)

Ecrire un programme qui calcule le nombre de manières que l'on peut faire la monnaie de 100€ en billets de 5, 10 et 20€. Le programme devra donner toutes les combinaisons possibles.

Exercice 5. (quoiquoi)

1) Que calcule la fonction ci-dessous ($j \geq 0$) . Écrire une fonction non récursive (avec une boucle) qui calcule le même résultat.

```
int quoiquoi (int i, int j)
{ if (j == 0) return 1;
  return i*quoiquoi(i,j-1);
}
```

Implémenter les deux versions d'algorithmes en Java.

2) Que calcule la fonction ci-dessous ? Simuler son exécution pour $n = 3$.

```
int quoiquoi (int n)
{ if (n == 0) return 0;
  return quoiquoi(n-1)+2*n-1;
}
```

Exercice 6. (pgcd)

Écrire une fonction récursive calculant le PGCD de deux entiers a et b . On suppose que le test $a > b$ est effectué avant l'appel de la fonction.

Exercice 7. (maximum - tableau)

Un tableau de taille N est composé d'un élément et d'un sous-tableau de taille $N - 1$. Écrire une fonction récursive calculant le maximum.

Exercice 8. (températures)

Écrire un programme qui trie les températures du mois de décembre à Belfort dans le sens ascendant.

Exercice 9. (tableaux 2D - matrices)

- i) Écrire un programme qui construit et affiche une matrice carrée unité U de dimension N .
Une matrice unité est une matrice, telle que : $U[i][j] = 1$ si $i = j$, 0 si $i \neq j$
- ii) Mise à zéro de la diagonale secondaire d'une matrice.
Écrire un programme qui saisit par lecture et affiche une matrice carrée A , puis met à zéro les éléments de la diagonale secondaire de A et affiche la matrice ainsi modifiée.