

Examen – INFORMATIQUE

Aucune sortie de la salle d'examen n'est autorisée durant la première heure de l'épreuve. Aucun document, ni téléphone, ni calculatrice n'est autorisé. Durée=1h30.

MAPLE

Seuls les opérateurs arithmétiques, la virgule, les parenthèses-crochets, l'affectation :=, les :, et les mots clés suivants sont autorisés : proc, return, end, if, and, or, not, then, fi, while, do, od, for, from, to, nops, NULL.

On attend les procédures les plus courtes possibles, et réutilisant au maximum les procédures des questions précédentes. La longueur des procédures sera pénalisée, typiquement 7-8 lignes est un maximum.

Exercice 1 Étant donnés deux entiers a et b , tels que $a \geq b$, il existe deux uniques entiers q (le quotient) et r (le reste) tels que $a = qb + r$ avec $0 \leq r < b$. Si le reste est nul, c'est-à-dire : $a = qb$, on dit que b est un diviseur de a . Clairement b est un diviseur de zéro (prendre $q = 0$). Clairement, 1 et a sont des diviseurs de a ; si a a exactement deux diviseurs, on dit que a est premier. On note $pgcd(a, b)$ le plus grand diviseur commun de a, b , par exemple $pgcd(21, 15) = 3$ ou $pgcd(15, 0) = 15$.

1) Écrire une fonction `IsDivisor:=proc(a,b)` prenant deux entiers a, b ($a \geq b$) en paramètre et qui renvoie vrai si b est un diviseur de a , faux sinon. Par-exemple :

```
> IsDivisor(17,5);  
false  
> IsDivisor(15,5);  
true
```

2) Écrire une fonction `DivisorS:=proc(x)` prenant un entier x en paramètre et retournant la liste L de ses diviseurs. Par-exemple :

```
> DivisorS(15);  
[1,3,5,15]  
> DivisorS(17);  
[1,17]
```

3) On dispose d'une fonction `member:=proc(L,x)` prenant une liste L d'entiers et un entier x en paramètre et retournant `true` si x est un élément de L , `false` sinon. Par-exemple :

```
> member([1,5,86,4,2],3);  
false  
> member([1,5,86,4,2],4);  
true
```

Écrire une fonction `Intersec:=proc(L1,L2)` prenant deux listes L_1, L_2 d'entiers triés dans l'ordre croissant en paramètre et retournant la liste des éléments communs aux deux listes dans l'ordre croissant.

```
> Intersec([1,3,4,5],[2,3,5,6]);
[3,5]
> Intersec([1,2,4,8,15],[2,3,4,15]);
[2,4,15]
```

4) Écrire une fonction `SumDivi:=proc(x)` prenant un entier x en paramètre et retournant la somme de ses diviseurs : Par-exemple :

```
> SumDivi(15);
24
> SumDivi(17);
18
```

5) Écrire une fonction `IsPrime:=proc(x)` prenant un entier x en paramètre et retournant vrai s'il est premier, faux sinon. Par-exemple :

```
> IsPrime(15);
false
> IsPrime(17);
true
```

6) Écrire une fonction `pgcd:=proc(a,b)` prenant deux entiers a, b ($a \geq b$) en paramètre et qui renvoie la valeur de $pgcd(a, b)$. Par-exemple :

```
> pgcd(21,15);
3
```

Écrire différemment la même fonction, que l'on appellera `pgcd2:=proc(a,b)`. La nouvelle fonction doit se baser sur l'égalité $pgcd(a, b) = pgcd(b, r)$.

EXCEL

Exercice 2 Le tableau Excel ci-dessous doit permettre l'évaluation d'un produit p en fonction de sa moyenne $m(p)$ sur deux indices /100. L'évaluation du produit est "A" s'il obtient un indice moyen $100 \leq m(p) < 70$, est "B" si $70 \leq m(p) < 50$, et "C" si $50 \leq m(p) \leq 0$.

	A	B	C	D	E
1	code produit	Indice1 (/100)	Indice2 (/100)	$m(p)$	évaluation
2	LMj	78	60		
3	gkG	28	25		
4	dfg	14	14		
5	fg	75	74		
6	eErtt	14	14		
7	rtYytr	12	45		

1) Donnez les formules contenues dans les cellules D2 et E2 pour qu'on puisse les étendre aux lignes 3 à 7. Par-exemple, LMj aura un $m(p) = 69$, et donc une évaluation B.

	F	G
1	Seuil1	Seuil2
2	70	50
3		
4		
5		
6		
7		

2) On souhaite ensuite avoir la possibilité de modifier les deux valeurs des seuils, 70 et 50, départageant les trois évaluations A,B,C, à d'autres valeurs. Par-exemple, si les seuils deviennent 65 et 45, l'évaluation de LMj devient A.

Faire les modifications nécessaires sur vos réponses à la première question pour permettre le changement des seuils.