AL 5I - Cryptographie - Contrôle continu numéro 1

Nom : _____

Prénom : _____

- Durée : 1 heure. Documents interdits, calculatrices interdites.
- Écrivez toutes les réponses directement sur le sujet. Si vous n'avez pas suffisamment de place, écrivez au dos d'une feuille en le **précisant dans la question**.

1 Question de cours (7 points)

1. [2 points] Expliquer avec un exemple ce qu'est un algorithme asymétrique.

   Solution: Si Alice souhaite que Bob lui envoie un message, il suffit de lui fournir une boîte munie d'un cadenas ouvert dont elle a conservé la clé. Bob n'a plus qu'à placer le message dans la boîte et verrouiller le cadenas. Seule la personne qui détient la clé, Alice, est maintenant capable de déchiffrer le message.

2. [2 points] Quels avantages offrent les algorithmes symétriques à clé par rapport aux algorithmes symétriques sans clé.

   Solution: L'efficacité d'un algorithme sans clé repose entièrement sur le secret de l'algorithme. Par conséquent, sa durée de vie est limitée et il dangereux de l'utiliser avec plusieurs correpondants différents.

3. [1 point] Donner la définition mathématique de $a|b$.

   Solution: $a|b \iff \exists k, ka = b$

4. [1 point] Définir mathématiquement ce qu'est un nombre premier. Vous préciserez les éventuels cas particuliers.

   Solution: $p > 1$ est premier s'il n'a que 1 et p comme diviseurs positifs. Par convention, 1 n'es pas premier.

5. [1 point] Qu'est que deux nombres premiers entre eux ?

Solution: Deux nombres sont premiers entre eux si leur *pgcd* est égal à 1.

2 Arithmétique (8 points)

1. [2 points] Effectuez les divisions suivantes :

a	b	q	r
27	18		
-14	5		
17	17		
8	13		

Solution:

a	b	q	r
27	18	1	9
-14	5	-3	1
17	17	1	0
8	13	0	8

2. [2 points] Donner les *pgcd* des couples de nombres suivants.

a	b	$pgcd(a,b)$
19	19	
35	49	
0	27	
26	65	

Solution:

a	b	$pgcd(a,b)$
19	19	19
35	49	7
0	27	27
26	65	13

3. [2 points] Compléter le tableau suivant :

a	décomposition de a
1	
80	
	$(0,0,\ldots)$
	$(0,4,0,\ldots)$

Solution:

a	décomposition de a
1	$(0,0,\ldots)$
80	$(4,0,1,0,\ldots)$
1	$(0,0,\ldots)$
81	$(0,4,0,\ldots)$

4. [2 points] Compléter le tableau suivant :

décomposition de a	décomposition de b	décomposition de $pgcd(a,b)$
$(1,3,2,0,1,0,\ldots)$	$(4,28,2,0,\ldots)$	
$(2,0,0,1,0\ldots)$	$(0,0,2,0,1,0,\ldots)$	
$(2,2,3,4,0\ldots)$	$(3,9,3,4,0,\ldots)$	
$(0,0,0,\ldots)$	$(0,2,1,0,\ldots)$	

Solution:

décomposition de a	décomposition de b	décomposition de $pgcd(a,b)$
$(1,3,2,0,1,0,\ldots)$	$(4,28,2,0,\ldots)$	$(1,3,2,0\ldots)$
$(2,0,0,1,0\ldots)$	$(0,0,2,0,1,0,\ldots)$	$(0,0,\ldots)$
$(2,2,3,4,0\ldots)$	$(3,9,3,4,0,\ldots)$	$(2,2,3,4,0,\ldots)$
$(0,0,0,\ldots)$	$(0,2,1,0,\ldots)$	$(0,0,\ldots)$

3 GMP (5 points)

1. 5 points Complétez le programme C suivant. Il doit afficher la somme des carrés des 100000 premiers entiers (c'est-à-dire $1^2 + 2^2 + \ldots + 99999^2 + 100000^2$). Vous utiliserez les fonctions suivantes :
   - `mpz_init(mpz_t x)`, initialise la variable x
   - `mpz_set_ui(mpz_t x, unsigned int i)`, affecte i à x
   - `mpz_mul_ui(mpz_t res, mpz_t y, unsigned int i)`, affecte à res le produit de y et de i.
   - `mpz_add(mpz_t res, mpz_t x, mpz_t y)`, affecte à res la somme de x et de y.
   - `mpz_out_str(FILE* f, int base, mpz_t x)`, exporte au format chaîne de caractère la variable x en base $base$ dans le flux f.

```
#include<stdio.h>
#include<gmp.h>

#define N 100000

int main()
{
   ..........  somme;

   ..........  carre;

   ..........  i;

   mpz_..........(somme);

   mpz_..........(carre);

   mpz_set_ui(..........., ...........);

   for(i = 1 ;i <= N ; i++)
     {
       mpz_set_ui(carre, i);

         ..........  ;

         ..........  ;
     }

   ..........;
   printf("\n");
```

```
  return 0;
}
```

Solution:

```
#include<stdio.h>
#include<gmp.h>

#define N 100000

int main()
{
  mpz_t somme;
  mpz_t carre;
  int i;
  mpz_init(somme);
  mpz_init(carre);
  mpz_set_ui(somme, 0);
  for(i = 1 ;i <= N ; i++)
    {
      mpz_set_ui(carre, i);
      mpz_mul_ui(carre, carre, i);
      mpz_add(somme, somme, carre);
    }
  mpz_out_str(NULL, 10, somme);
  mpz_clear(somme);
  mpz_clear(carre);
  printf("\n");
  return 0;
}
```