(1) Convertir 127_{10} et 93_{10} en binaire (en passant pas la base 8 + conversion rapide), les additionner dans cette représentation. Convertir le résultat en décimal pour vérifier qu'on n'a pas fait d'erreur.

```
127 | 8
    |--------
 47 |  15 | 8
  7 |     |----
    |   7 | 1
```

```
93 | 8
   |--------
13 |  11 | 8
 5 |     |----
   |   3 | 1
```

127_{10} = 7 + 8 * (7 + 8 * (1))
= 177_{8}
= 1111111_{2}

93_{10} = 5 + 8 * (3 + 8 * (1))
= 135_{8}
= 1011101_{2}

```
  rrrrrrr
   1111111
 + 1011101
-----------
  11011100
```

11011100_{2} = 334_{8}
= 4 + 8 * (3 + 8 * (3))
= 4 + 8 * (3 + 24)
= 4 + 8 * (27)
= 4 + 8 * (20 + 7)
= 4 + 160 + 56
= 220_{10}

(2) Multiplier 35_{10} et 211_{10} avec l'algorithme de multiplication sans table.

on met 1 et 211 dans la première ligne d'une table, on additionne chaque ligne à elle-même, jusqu'à ce que la première case dépasse le multiplicateur :

1	211
2	422
4	844
8	1688
16	3376
32	6752
64	...

On additionne les lignes qui permettent d'obtenir le multiplicateur (35)
1 + 2 + 32 = 35
211 + 422 + 6752 = 7385
donc **35_{10} × 211_{10} = 7385_{10}**

(3) Convertir 35_{10} et 211_{10} en binaire et faire tourner à la main l'algorithme de multiplication (convertir le résultat en décimal pour le vérifier).

```
35 | 8
   |----
 3 | 4
```

35_{10} = 3 + 8 * (4)
= 43_{8}
= 100011_{2}

```
211 | 8
    |--------
 51 |  26 | 8
  3 |     |----
    |   2 | 3
```

211_{10} = 3 + 8 * (2 + 8 * (3))
= 323_{8}
= 011010011_{2}

Algorithme pour multiplier a et b :

```
r ← 0
tant que a est différent de 0
      si le bit de droite de $a$ vaut 1
            r ← r + b
      décaler a droite
      décaler b gauche
```

En le faisant tourner à la main :
a = 100011
b = 11010011
r = 0

a != 0 :
- le bit à droite de a vaut 1 :
  - r = r + b = 11010011
- a = 10001
- b = 110100110

a != 0 :
- le bit à droite de a vaut 1 :
  - r = r + b =

```
         r    rr
          11010011
       + 110100110
      ------------
        1001111001
```

- a = 1000
- b = 1101001100

a != 0 :
- le bit à droite de a vaut 0
- a = 100
- b = 11010011000

a != 0 :
- le bit à droite de a vaut 0
- a = 10
- b = 110100110000

a != 0 :
- le bit à droite de a vaut 0
- a = 1
- b = 1101001100000

a != 0 :
- le bit à droite de a vaut 1 :
  - r = r + b =

```
        rr  rr
          1001111001
     + 1101001100000
     ---------------
       1110011011001
```

- a =
- b = 1000010011000000

terminé !

$1110011011001_2 = 16331_8$
$= 1 + 8 * (3 + 8 * (3 + 8 * (6 + 8 * (1))))$
$= 1 + 8 * (3 + 8 * (3 + 8 * (14)))$
$= 1 + 8 * (3 + 8 * (3 + 8 * (10 + 4)))$
$= 1 + 8 * (3 + 8 * (3 + 80 + 32))$
$= 1 + 8 * (3 + 8 * (115))$
$= 1 + 8 * (3 + 8 * (100 + 10 + 5))$
$= 1 + 8 * (3 + 800 + 80 + 40)$
$= 1 + 8 * (923)$
$= 1 + 8 * (900 + 20 + 3)$
$= 1 + 7200 + 160 + 24$
$= 7385_{10}$

(4) Poser la multiplication de 11001_2 et 101110_2

```
       101110
     *  11001
-------------
     rrrr
       101110
      000000.
     000000..
    101110...
   101110....
-------------
  10001111110
```

Vérification :

$101110_2 = 56_8$

$= 6 + 8 * (5)$

$= 46_{10}$

$11001_2 = 31_8$

$= 1 + 8 * (3)$

$= 25_{10}$

$10001111110_2 = 2176_8$

$= 6 + 8 * (7 + 8 * (1 + 8 * (2)))$

$= 6 + 8 * (7 + 8 * (17))$

$= 6 + 8 * (7 + 8 * (10 + 7))$

$= 6 + 8 * (7 + 80 + 56)$

$= 6 + 8 * (143)$

$= 6 + 8 * (100 + 40 + 3)$

$= 6 + 800 + 320 + 24$

$= 1150_{10}$