

```

pour partie1 :lettre
donne "code "
si (( :lettre = "a ) ou ( :lettre = "A )) donne "code "AF
si (( :lettre = "b ) ou ( :lettre = "B )) donne "code "XX
si (( :lettre = "c ) ou ( :lettre = "C )) donne "code "DD
si (( :lettre = "d ) ou ( :lettre = "D )) donne "code "VD
si (( :lettre = "e ) ou ( :lettre = "E )) donne "code "AX
si (( :lettre = "f ) ou ( :lettre = "F )) donne "code "FA
si (( :lettre = "g ) ou ( :lettre = "G )) donne "code "GG
si (( :lettre = "h ) ou ( :lettre = "H )) donne "code "DV
si (( :lettre = "i ) ou ( :lettre = "I )) donne "code "GD
si (( :lettre = "j ) ou ( :lettre = "J )) donne "code "XF
si (( :lettre = "k ) ou ( :lettre = "K )) donne "code "GX
si (( :lettre = "l ) ou ( :lettre = "L )) donne "code "AG
si (( :lettre = "m ) ou ( :lettre = "M )) donne "code "FG
si (( :lettre = "n ) ou ( :lettre = "N )) donne "code "VV
si (( :lettre = "o ) ou ( :lettre = "O )) donne "code "DX
si (( :lettre = "p ) ou ( :lettre = "P )) donne "code "VA
si (( :lettre = "q ) ou ( :lettre = "Q )) donne "code "AA
si (( :lettre = "r ) ou ( :lettre = "R )) donne "code "DF
si (( :lettre = "s ) ou ( :lettre = "S )) donne "code "AV
si (( :lettre = "t ) ou ( :lettre = "T )) donne "code "GF
si (( :lettre = "u ) ou ( :lettre = "U )) donne "code "GV
si (( :lettre = "v ) ou ( :lettre = "V )) donne "code "VX
si (( :lettre = "w ) ou ( :lettre = "W )) donne "code "XV
si (( :lettre = "x ) ou ( :lettre = "X )) donne "code "DG
si (( :lettre = "y ) ou ( :lettre = "Y )) donne "code "AD
si (( :lettre = "z ) ou ( :lettre = "Z )) donne "code "DA
si ( :lettre = "0 ) donne "code "FD
si ( :lettre = "1 ) donne "code "XA
si ( :lettre = "2 ) donne "code "VG
si ( :lettre = "3 ) donne "code "GA
si ( :lettre = "4 ) donne "code "FF
si ( :lettre = "5 ) donne "code "XD
si ( :lettre = "6 ) donne "code "VF
si ( :lettre = "7 ) donne "code "FX
si ( :lettre = "8 ) donne "code "FV
si ( :lettre = "9 ) donne "code "XG
rends :code
fin

```

```

pour adfgvx
efftxt partage [ "clé "tableau ]
ecl [ Tape ton message: ]
donne "message lisliste
ecl :message
donne "mess []
répète card :message donne "mess ph :mess sépare item :message boucle
donne "message :mess donne "mess "
répète card :message donne "mess mot :mess partie1 item :message boucle
donne "message :mess
ecl [ Tape la clé de codage: ]
donne "clé lisliste
ecl :clé ecl "
donne "clé2 []
répète card :clé donne "clé2 ph :clé2 sépare item :clé boucle
donne "clé :clé2
donne "clé2 []
répète card :clé donne "clé2 ph :clé2 0
donne "tableau []
répète card :clé donne "tableau liste :tableau :clé2
donne "tableau sp :tableau
donne "mess []
répète card :message donne "mess ph :mess sépare item :message boucle
répète card :clé [
  donne "b1 boucle donne "b2 :b1
  répète card :clé [
    teste (:b2 <= card :mess)
    sivrai remplace :tableau item :mess :b2 ph :b1 boucle
    sifaux remplace :tableau [*] ph :b1 boucle
    donne "b2 :b2 + card :clé
  ]
]
répète card :tableau ecl item :tableau boucle
répète card :clé remplace :clé ascii item :clé boucle ph boucle []
ordonneliste donne "code "

```

```

répète card :clé [
    donne "b1 boucle
    répète card :clé [
        donne "atome saisis :tableau ph boucle :b1
        teste :atome = [*]
        sivrai donne "code mot :code car 32
        sifaux donne "code mot :code :atome
    ]
]
ecl :code
stop
fin

adfgvx

pour EchangeItems :id
    donne "a item :clé :id donne "b item :clé (:id + 1)
    donne "c item :tableau :id donne "d item :tableau (:id + 1)
    remplace :clé :b ph :id []
    remplace :clé :a ph (:id + 1) []
    remplace :tableau :d ph :id []
    remplace :tableau :c ph (:id + 1) []
fin

pour compare :id :o
    teste (( item :clé :id ) < ( item :clé ( :id + 1 )))
    sivrai rends :o
    sifaux [
        EchangeItems :id
        rends :id +1
    ]
fin

pour ordonneliste
    donne "ki 1 donne "ks card :clé
    tantque ( :ki < :ks ) [
        donne "nks :ki donne "nki :ks
        donne "q 0
        répète ( :ks - :ki ) [
            donne "nks compare :ki + :q :nks
            donne "nki compare ( :ks - :q -1) :nki
            donne "q :q + 1
        ]
        donne "ki :nki donne "ks :nks
    ]
fin

```