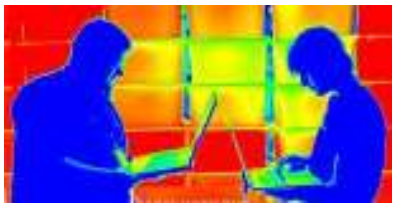
12 Programul final CRYPTO_CAESAR

```
def crypto_in(mess,key):
  KEY = key
  tx_in= mess
  tx_out=""

  for litera in tx_in:
    litera_noua = litera
    if litera.isalpha():
      num = ord(litera)
      num = num + KEY
      if litera.isupper() and num > ord("Z"):
        num = num - 26
      if litera.islower() and num > ord("z"):
        num = num - 26
      litera_noua = chr(num)

    tx_out=tx_out + litera_noua
  return tx_out

def crypto_out(mess,key):
  KEY = key
  tx_in= mess
  tx_out=""
  for litera in tx_in:
    litera_noua = litera
    if litera.isalpha():
      num = ord(litera)
      num = num - KEY
      if litera.isupper() and num < ord("A"):
        num = num + 26
      if litera.islower() and num < ord("a"):
        num = num + 26
      litera_noua = chr(num)

    tx_out=tx_out + litera_noua
  return tx_out

mod = input(" Doresti sa faci o Criptare(C) sau o Decriptare (D)?")
mod = mod.upper()

mesaj  = input("Introdu textul tau:")
```

… continuare

```
ok = False
while not ok:
  ok = True
  try:
    vKey = int(input("Cheia de criptare este:   "))
    if (vKey not in range(1,27)):
        print("Cheia trebuie sa fie un nr intre 1 si 26 !")
        ok = False
  except ValueError:
    print("Cheia trebuie sa fie un numar ! ")
    ok = False

if mod == "C":
  mesaj2 = crypto_in(mesaj,vKey)
  print("Text criptat:   ", mesaj2)
elif mod == "D":
  mesaj2 = crypto_out(mesaj,vKey)
  print("Text decriptat: ", mesaj2)
else:
  print(" Va multumesc!  ")
```

13 Bonus! Spargerea codului!



Cum putem sparge un mesaj criptat?
Având un calculator la dispoziţie si cunoscând programare Python, nici nu e chiat atât de greu!
Să presupunem că am interceptat un mesaj criptat despre care nu ştim decât că e criptat cu metoda CAESAR (prin decalarea literelor, dar fară modificarea caracterelor speciale). Desigur, nu cunoaştem nici cheia de criptare.
Vom folosi metoda forţei brute, adică vom scrie un program care va încerca toate cheile posibile (de la 1 la 26) vom verifica rezultatele afişate şi îl vom alege pe cel care e “citibil”.

Adaugă la programul tău funcţia spargere:

```
def spargere(mess):
   for i in range(1,27):
       mes2=crypto_out(mess,i)
       print("Varianta " +str(i) + " =  " + mes2 )
```

Ai observat cum apelăm funcţia **crypto_out** din interiorul acestei functii? Şi încă de mai multe ori!

Modifică secvenţa cu modul de operare astfel:

```
mod = input(" Doresti sa faci o Criptare(C),  o Decriptare (D) sau o spargere (S) ? ")
mod = mod.upper()
```

Secvenţa de cod prin care se cere cheia, o ivom ntroduce într-o funcţe numită cere_cheia(): si o vom apela numai din modurile C sau D. Atunci secvenţa de la final se modifica astfel:

```
if mod == "C":
  vKey = cere_cheia()
   mesaj2 = crypto_in(mesaj,vKey)
   print("Text criptat:   ", mesaj2)
elif mod == "D":
     vKey = cere_cheia()
   mesaj2 = crypto_out(mesaj,vKey)
   print("Text decriptat: ", mesaj2)
elif mod == "S":
   spargere(mesaj)
else:
   print(" Va multumesc!  ")
```

Am adăugat o nouă opţiune la structura IF, pentru modul “S” , care va apela funcţia “spargere” Printarea are loc acolo, în funcţie, deci nu mai e nevoie de ea aici.

14 Şi acum testarea finală!

Execută programul tău cu opţiunea “C”.
Introdu textul “Azi e o zi frumoasa”, cu cheia de cripatre 7. Copiază rezultatul obţinut cu CTRL+C şi trimite mesajul criptat prietenului tău (prin email, prin facebook, cum doreşti).
Prietenul tău, dacă are acest program Python la dispozitie, va decripta mesajul astfel:
Îl copiază cu CTRL+C, porneste programul Python, alege opţiunea “D” şi dă paste cu CTRL+V şi va putea citi mesajul tău decriptat.
Acum, încearcă o “spargere”. Roagă-l pe prietenul tău să îţi trimită un alt mesaj criptat cu o cheie nouă, pe care nu o cunoşti şi încearcă să îl decriptezi cu opţiunea “S”.

Felicitări ! Ai scris primul tău proiect