Chelnokov Nikita, N41534, MFMC 1.1, HW3 pt. 1

Task 1 | (3pts)

(a) p - prime

Prove $|Q_p| = |\bar{Q}_p| = \frac{p-1}{2}$. There are exactly $\frac{p-1}{2}$ residues in Z_p^*, congruent with: $1^2, 2^2, \dots, (\frac{p-1}{2})^2$. Proovement:

1) $x^2 \equiv (p-x)^2 \bmod p \iff \cancel{x^2} - p^2 + 2px - \cancel{x^2} \vdots p \iff p(2x-p) \vdots p$

2) Let $x \neq p-y$, $x \neq y$, $0 \le x, y \le \frac{p-1}{2}$

$\nexists\ x^2 \equiv (p-y)^2 \bmod p \Rightarrow x^2 - (p-y)^2 \vdots p \iff x^2 - p^2 + 2py - y^2 \vdots p \Rightarrow$

$\Rightarrow \left.\begin{matrix} x^2 - y^2 \vdots p \\ p\text{-prime} \end{matrix}\right] \Rightarrow \left[\begin{matrix} x+y \vdots p \text{ - contradiction } (x+y \le p-1) \\ x-y \vdots p \text{ - contradiction } (|x-y| < p) \end{matrix}\right.$

1 and 2 $\Rightarrow$ every $(x, p-x)^*$ has a specific residue, while there are no residues in $(x, p-y)$, where $x \neq y$. Statement proven.

$^* x \in [1, 2, \frac{p-1}{2}]$

(b) p - prime

Let $b^2 \overset{*}{\equiv} c^2 \overset{\#}{\equiv} a \bmod p$, $b \neq c$, $b, c < p$

$\left.\begin{matrix} b^2 - c^2 \vdots p = (b-c)(b+c) \vdots p \\ p\text{-prime} \end{matrix}\right] \Rightarrow \left[\begin{matrix} b - c \vdots p \Rightarrow b \equiv c \bmod p \ (1) \\ b + c \vdots p \Rightarrow b \equiv (-c) \bmod p \ (2) \end{matrix}\right.$

(1) & (2) are wrong (*, #, p-prime) $\Rightarrow$ if b is a solution, then the other solution can only be $-b$. Statement proven

Task 2 | (3 pts)

(a) 1) $\left(\frac{8}{13}\right) = \left(-\frac{4}{13}\right) = \frac{2}{13} = -\frac{1}{13} = -1$

2) $\left(\frac{9}{13}\right) = \left(\frac{4}{9}\right) = \frac{2}{9} = \frac{1}{9} = 1$

3) $\left(\frac{14}{21}\right) = \left(-\frac{7}{21}\right) = \left(-\frac{0}{7}\right) = 0$

4) $\left(\frac{15}{21}\right) = \left(\frac{6}{15}\right) = \left(\frac{3}{15}\right) = \left(-\frac{0}{3}\right) = 0$

5) $\left(\frac{100}{100}\right) = \left(\frac{100}{5}\right)^2 \cdot \left(\frac{100}{2}\right)^2 = 0$

6) $\left(\frac{290}{431}\right) = \left(\frac{145}{431}\right) = \left(\frac{141}{145}\right) = \left(\frac{4}{141}\right) = \left(-\frac{2}{141}\right) = \frac{1}{141} = 1$

(b) $n = 5 \cdot 7 = 35$

1) residues = 0, 1, 4, 9, 11, 14, 15, 16, 21, 25, 29, 30

nonresidues = 2, 3, 5, 6, 7, 8, 10, 12, 13, 17, 18, 19, 20, 22, 23, 24, 26, 27, 28, 31, 32, 33, 34 // I coded algo to find residues (ref. Task 1)

2,3)

	$\left(\frac{a}{5}\right)$	$\left(\frac{a}{7}\right)$	$\left(\frac{a}{35}\right)$
1	1	1	1
2	-1	1	-1
3	-1	-1	1
4	1	1	1
5	0	-1	0
6	1	-1	-1
7	-1	0	0
8	-1	1	-1
9	1	1	1
10	0	-1	0
11	1	1	1
12	-1	-1	1
13	-1	-1	1
14	1	0	0
15	0	1	0
16	1	1	1
17	-1	-1	1
18	-1	1	-1
19	1	-1	-1
20	0	-1	0
21	1	0	0
22	-1	1	-1
23	-1	1	-1
24	1	-1	-1
25	0	1	0
26	1	-1	-1
27	-1	-1	1
28	-1	0	0
29	1	1	1
30	0	1	0
31	1	-1	-1
32	-1	1	-1
33	-1	-1	1
34	1	-1	-1
35	0	0	0

4) 3, 12, 13, 17, 27, 33

Task 3 | (4 pts)

The message M is: Turing

Code for solution provided with the pdf.

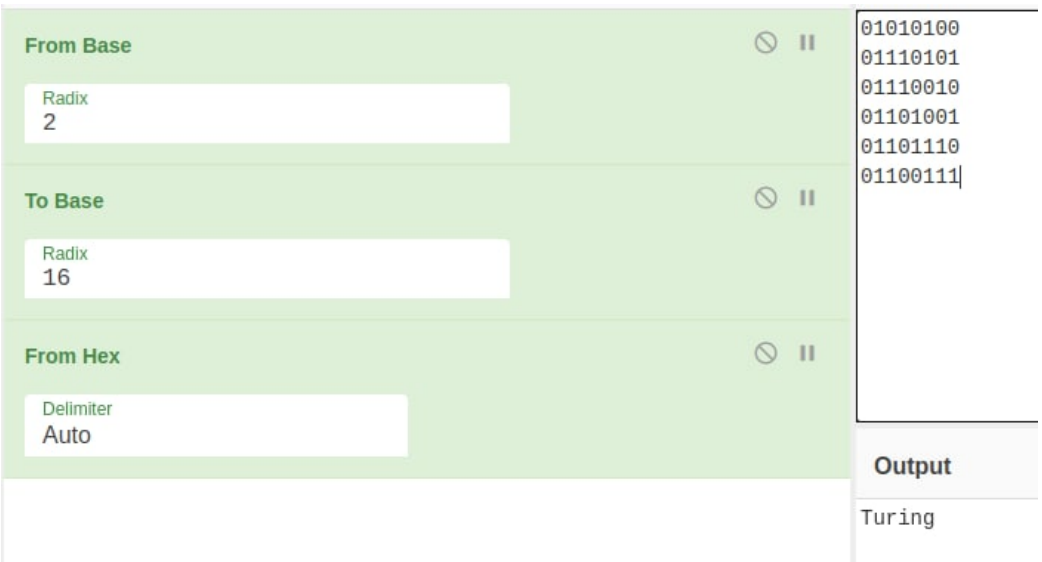
Task 4 | (6 pts)

Code provided with the pdf.

Task 3 details:

```
def dec(encarr, p=13, q=19):
  res = []
  for c in encarr:
    c = c % p
    residues = list(set([i**2 % p for i in range(p)]))
    nonresidues = set(range(p)).difference(residues)
    if c in residues:
      res.append('0')
    else:
      res.append('1')
  for pt in res:
    print(pt, end='')
  print()
  return

ena = [218, 34, 194, 164, 220, 50, 237, 77]
dec(ena)
ena = [68, 151, 135, 21, 101, 167, 196, 98]
dec(ena)
ena = [196, 219, 89, 241, 16, 134, 240, 43]
dec(ena)
ena = [36, 193, 37, 17, 184, 61, 81, 41]
dec(ena)
ena = [81, 148, 18, 172, 193, 37, 203, 233]
dec(ena)
ena = [244, 145, 18, 1, 121, 46, 18, 193]
dec(ena)
```



Task 4 details:

a)

```
from random import *

def fermprim(n, k=10):
    if (n == 2):
        print('Prime')
        return
    elif (n % 2 == 0):
        print('Composite')
        return
    else:
        for i in range(k):
            a = randint(1, n-1)
            if (pow(a, n-1, n) != 1):
                print('Composite')
                return
        print('Prime')
    return

print('Number 561 is: ')
fermprim(561, 10)
print('Number 4241 is: ')
fermprim(4241, 100)
print('Number 25879203 is: ')
fermprim(25879203)
```

```
skifast1@BlackFlux 〉 ~/Downloads/CR  python3 fermprim.py
Number 561 is:
Composite
Number 4241 is:
Prime
Number 25879203 is:
Composite
```

b)

```
from random import *

def jacobiCalc(a, n):
    a = a % n
    res = 1
    while (a != 0):
        while (a % 2 == 0):
            a = a // 2
            tarr = [3, 5]
            if ((n % 8) in tarr):
                res = res*-1
        a, n = n, a
        if (a % 4 == n % 4 == 3):
            res = res*-1
        a = a % n
    if (n == 1):
        return res
    return 0
```

```
def solostrassCalc(p, k=100):
    if (p == 2):
        print('Prime')
        return
    elif (p % 2 == 0):
        print('Composite')
        return
    for i in range(k):
        a = randint(1, p-1)
        jacobi_sym = (p + jacobiCalc(a, p)) % p
        e = (p - 1) // 2
        res = pow(a, e, p)
        if (jacobi_sym == 0 or res != jacobi_sym):
            print('Composite')
            return
    print('Prime')
    return

print('Number 561 is: ')
solostrassCalc(561)
print('Number 4241 is: ')
solostrassCalc(4241)
print('Number 25879203 is: ')
solostrassCalc(25879203)
```

```
skifast1@BlackFlux 〉 ~/Downloads/CR  python3 solostrass.py
Number 561 is:
Composite
Number 4241 is:
Prime
Number 25879203 is:
Composite
```