Name:

Math 424: Exam 1

Make sure to show all your work as clearly as possible. This includes justifying your answers if required. Calculators are not allowed.

You may use any result from the chapters covered in the text or from lecture. You may not use the results of homework or worksheet problems.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

1. Short answer questions.

   (a) (10 pts) Compute $\gcd(286, 299)$.

   Solution: $299 = 1 \cdot 286 + 13$ and $286 = 22 \cdot 13 + 0$, so $\gcd(286, 299) = 13$.

   (b) (10 pts) Compute the inverse of 11 mod 79.

   Solution: Our table for the Extended Euclidean algorithm is

r	q	s	t
79	–	1	0
11	–	0	1
2	7	1	-7
1	5	-5	36
0			

   Thus $-5 \cdot 79 + 36 \cdot 11 = 1$; taking this mod 79 yields $36 \cdot 11 \equiv 1 \pmod{79}$. The inverse is therefore 36.

   (c) (10 pts) Convert 53 to binary.

   Solution: A routine calculation yields 110101_2.

   (d) (10 pts) Suppose $n \in \mathbb{N}$ satisfies $16 \mid n$. What can you say about the binary expansion of n?

   Solution: The last 4 bits are 0: if the binary expansion is $(d_{m-1}d_{m-2}\ldots d_3d_2d_1d_0)_2$, then we have

   $$n = \sum_{i=0}^{m-1} d_i 2^i.$$

   Taking both sides mod 16, and observing that $2^i \equiv 0 \pmod{16}$ whenever $i \geq 4$, we get

   $$0 \equiv d_3 \cdot 2^3 + d_2 \cdot 2^2 + d_1 \cdot 2^1 + d_0 \cdot 2^0 \pmod{16}.$$

   Since the right side is < 16, in fact

   $$d_3 \cdot 2^3 + d_2 \cdot 2^2 + d_1 \cdot 2^1 + d_0 \cdot 2^0 = 0.$$

   As $d_i = 0$ or 1, it is not hard to check that we must have $d_i = 0$ for $0 \leq i \leq 3$.

2. (a) (10 pts) The following Python function takes as input two positive integers a and b:

```
def cat(a,b):
    while a > b:
        a = a - b
    if a == b:
        a = 0
    return a
```

What does the function do? Give a brief informal explanation.

Solution: It computes the remainder of a on division by b. By the division algorithm, $a = qb+r$ for some $0 \leq r < b$. The algorithm replaces this with $(q-1)b+r$ on each iteration, until we are left with either r (if $r > 0$) or b (if $r = 0$. In the former case, the algorithm outputs r; in the latter, 0. The claim follows.

(b) (10 pts) The following Python function takes as input two positive integers a and n.

```
def dog(a, n):
    for i in range(n):
        if (a*i - 1)%n == 0:
            return i
    return False
```

What does the function do? Give a brief informal explanation.

Solution: It computes the inverse of a mod n, if it exists, and outputs `False` if it doesn't. If the inverse i does exist, then we can take $0 \leq i \leq n-1$, and $ai \equiv 1 \pmod{n}$. This last is equivalent to $n \mid (ai-1)$, which is exactly the return condition in the loop.

3. (15 pts) Alice sends Bob the message "GRIKP" encrypted with a Caesar cipher. Bob cannot remember the key, but he *does* remember that "BOB" encrypts to "SFS". What was Alice's message?

Solution: As $B = 1$ and $S = 18$, we have $k = 18 - 1 = 17$. Subtract 17 from each letter of the ciphertext yields the message "PARTY".

4. The Klingon alphabet consists of 44 characters; we will use the numbers 0 through 43 to represent them. The Klingons wish to use an affine cipher to encrypt their messages.

   (a) (5 pts) If the cipher is given by $E(P,(a,b)) = aP + b \pmod{44}$, what conditions must a and b satisfy?

   Solution: We need $\gcd(a, 44) = 1$ so that encryption is invertible. There is no condition on b.

   (b) (15 pts) A Klingon message has been encrypted with the key $(19, 2)$. The ciphertext is $[5, 9, 12]$. What is the plaintext?

   Solution: We need to compute $19^{-1} \pmod{44}$. The Extended Euclidean algorithm gives the following table:

r	q	s	t
44	–	1	0
19	–	0	1
6	2	1	-2
1	3	-3	7
0			

   This gives $-3 \cdot 44 + 7 \cdot 19 = 1$, so $7 \cdot 19 \equiv 1 \pmod{44}$. Now we apply $D(C,(19,2)) = 7 \cdot (C-2)$

(mod 44) to get

$$P = [7 \cdot (5-2) \pmod{44}, 7 \cdot (9-2) \pmod{44}, 7 \cdot (12-2) \pmod{44}]$$
$$= [21, 5, 26].$$

5. (15 pts) Suppose that $a, b, d \in \mathbb{Z}$ with $\gcd(a, 26) = d$ and $d \neq 1$. Prove that if we encrypt with the affine cipher $E(P, (a, b)) = aP + b \pmod{26}$, then there are two plaintexts which encrypt to the same ciphertext.

Solution: Observe that $E(\text{“A”}, (a, b)) = a \cdot 0 + b \pmod{26} = b \pmod{26}$. We will find another letter that encrypts to the same thing.

Since $d \mid a$, $a = cd$ for some $c \in \mathbb{Z}$. Since $d \mid 26$, we have $d = 26, 13$, or 2. If $d = 26$, then $a = 26c \equiv 0 \pmod{26}$, and so the encryptions of A and B (and in fact of *every* letter) are both $b \pmod{26}$.

If $d = 13$, then

$$E(\text{“C”}, (a, b)) = a \cdot 2 + b \pmod{26} = 26c + b \equiv b \pmod{26}.$$

Finally, if $d = 2$, then

$$E(\text{“N”}, (a, b)) = a \cdot 13 + b \pmod{26} = 26c + b \equiv b \pmod{26}.$$

6. (15 pts) Suppose we use a language that has only 3 letters, a, b, and c. The frequency of these characters in the language is 0.8, 0.15, and 0.05, respectively. The ciphertext “ABCBABBBAC” was encrypted with a Vigenère cipher with key length 2. What is the most likely plaintext?

Solution: The two slices are “ACABA” and “BBBBC”, respectively.

In the first slice, “A” is the most common, so likely stands for “a”. Thus the first shift is 0. The first slice decrypts to “acaba”.

In the second slice, “B” is the most common, so likely stands for “a”. Thus the second shift is 1. The second slice decrypts to “aaaab”.

Putting the slices back together yields “aacaaabaab”.