1. One day while walking around the St. Olaf College campus you come to a branch in your path, where you must turn left or right. One direction leads to Rolvaag Library (where you want to go) and the other to the High-Voltage Pain Complex (which you'd rather avoid). A student offers to help. From reading Wikipedia you know that there are two kinds of St. Olaf students: those who always tell the truth, and those who always lie. Given that fact, what single "yes or no" question can you ask the student, to determine immediately which direction leads to the library? Explain.

Answer: "If I were to ask you, 'Does the left path go to the library?', would you answer, 'Yes'?" Then, if the student answers "Yes", then go left; if the student answers "No", then go right. [You can see that this works by analyzing four cases, based on whether (1) the left path goes to the library and (2) whether the student is a truth-teller. I will leave this work to you. Also, this solution is not unique; there may be many questions that accomplish the key goal of simultaneously probing the student and the path.]

2. Prove that every odd integer is the difference of two squares.

Answer: Let n be any odd integer. Then by the definition of *odd* there exists an integer k such that $n = 2k + 1$. Notice that

$$(k+1)^2 - k^2 = k^2 + 2k + 1 - k^2 = 2k + 1 = n.$$

Therefore n is the difference of the squares $(k + 1)^2$ and k^2, as desired.

3. Each part of this problem is a true/false question, but there are three valid answers: TRUE, FALSE, and PUNT. If you answer PUNT, then you receive half credit. Otherwise, if you answer correctly then you receive full credit, and if you answer incorrectly then you receive no credit. No explanation is necessary on this problem. Do not just write T, F, or P; write the entire word, clearly. Parts A-D are about the integers; parts E-H are about logic.

A. $\forall a\, \forall b\, \forall c\, \forall m\, (ac \equiv bc \pmod{m} \rightarrow a \equiv b \pmod{m})$

Answer: FALSE [For example, $a = 3$, $b = 6$, $c = 4$, $m = 12$.]

B. $\forall a\, \forall b\, \forall c\, \forall m\, (a \equiv b \pmod{m} \rightarrow ac \equiv bc \pmod{m})$

Answer: TRUE [This follows from our basic lemma of modular arithmetic, or equivalently Theorem 5 in Section 3.4.]

C. $\forall a\, \forall b\, \forall c\, (a|bc \rightarrow (a|b \vee a|c))$.

Answer: FALSE [This is true if and only if the variable a is restricted to primes.]

D. $\forall a\, \forall m\, \exists b\; ab \equiv \gcd(a, m) \pmod{m}$.

Answer: TRUE [This follows from Theorem 1 in Section 3.7.]

E. The truth table for the compound proposition $(p \rightarrow q) \vee (r \rightarrow \neg q)$ has exactly 6 true rows.

Answer: FALSE [It has 8 true rows.]

F. $\forall x\, \forall y\; p(x, y)$ is logically equivalent to $\forall y\, \forall x\; p(x, y)$ for all predicates $p(x, y)$ and all domains.

Answer: TRUE [You can reorder quantifiers as long as they are of the same type (existential or universal).]

G. $\neg\forall x\ (p(x) \wedge q(x))$ is logically equivalent to $\exists x\ (\neg p(x) \wedge \neg q(x))$ for all predicates $p(x)$ and $q(x)$ and all domains.

Answer: FALSE [In the latter expression, the $\wedge$ should be a $\vee$, by DeMorgan's Law.]

H. Let p be the statement "Elvis is 6 years old" and q the statement "Elvis enjoys Disney World." Then the statement "Elvis enjoys Disney World only if Elvis is 6 years old" is $p \to q$.

Answer: FALSE [It's $q \to p$.]

4. Compute the inverse of 27 modulo 140. That is, find an integer y such that $0 \le y < 140$ and $27y \bmod 140 = 1$.

Answer: [Here is an outline. I leave the details to you.] Use the Extended Euclidean Algorithm (page 232) to find numbers s and t such that

$$\gcd(140, 27) = s \cdot 140 + t \cdot 27.$$

You will probably find $s = 11$ and $t = -57$. The inverse is then $t \bmod 140 = 83$.

5. This is a 3-part question about RSA. If you can't answer one part, then go on to the next.

A. Let p and q be distinct primes. Prove that $\phi(pq) = (p-1)(q-1)$.

Answer: First, $\phi(pq)$ is the number of integers in $\{1, \ldots, pq\}$ that are relatively prime to pq. For an integer in $\{1, \ldots, pq\}$ *not* to be relatively prime to pq, it must share a factor with pq. The only factors of pq are p and q. So the integers in $\{1, \ldots, pq\}$ that are not relatively prime to pq are exactly those that are multiples of p or q. The multiples of p are

$$p, 2p, 3p, \ldots, qp$$

(there are q of them) and the multiples of q are

$$q, 2q, 3q, \ldots, pq$$

(there are p of them). These lists do not overlap, except at pq itself. So there are $q + p - 1$ numbers in $\{1, \ldots, pq\}$ that are not relatively prime to pq. This means that

$$\phi(pq) = pq - (q + p - 1) = (p-1)(q-1).$$

B. Suppose that you've discovered a fast algorithm for computing $\phi(k)$ for any positive integer k. Explain how you could use this to read anyone's RSA-encrypted messages.

Answer: To read someone's RSA-encrypted messages, I first obtain their modulus n and encryption key e, which are public. Then I compute $\phi(n)$. Since $n = pq$ for the person's chosen primes p and q, I know now $\phi(n) = (p-1)(q-1)$, by Part A of this problem. The person's decryption key d is simply the inverse of e modulo $(p-1)(q-1)$; just as easily as that person computed d, I can compute d too (using the Extended Euclidean Algorithm, page 232). Thus I can decrypt their messages just as easily as they can.

C. Suppose instead that you've discovered a fast algorithm for factoring integers. Explain how you could use this to read RSA-encrypted messages.

Answer: To read someone's RSA-encrypted messages, I first obtain their modulus n and encryption key e, which are public. Then I factor n into $n = pq$, recovering the primes p and q that the person began with. Then I compute $(p-1)(q-1)$; from this and the encryption key e I can compute the decryption key d, just as in Part B.

6. The (n, k) Hamming code encodes k bits into n bits by adding $n-k$ extra bits for error correction. Our example from class used $n = 7$ and $k = 4$ — that is, 3 extra bits. For your reference, the matrices we used are given below. Now, I'm interested in the "next largest" Hamming code — one that uses 4 extra bits to handle errors, instead of 3.

$$E = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A. How big can k and n be, if 4 error-correction bits are used? Explain, of course.

Answer: First, E must encode k bits into n, so it must be $n \times k$. Similarly, D must decode n bits into k, so it must be $k \times n$. A key property is that the columns of C give the numbers $1, 2, \ldots, n$ in binary. Therefore $n = 2^b - 1$, where b is the number of rows in C. It turns out that $b = n - k$ works; that is, the number of rows in C equals the number of error-correction bits. So for $b = n - k = 4$ we get $n = 15$ and $k = 11$.

B. Write down D, C, and E matrices for this "next largest" Hamming code. (Note: E is not worth many points, so omit E if you don't have time.)

Answer: It is perhaps easiest to begin with

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Notice in the matrix D above that columns 1, 2, and 4 are all 0s, and that the other columns form the identity matrix I_4. Continuing this pattern, our new D has 0s in columns 1, 2, 4, and

8 and the identity matrix I_{11} elsewhere:

$$D = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Finally, in the matrix E above notice that rows 1, 2, and 4 are strange, and that the other rows form I_4. (This is responsible for the key property that $DE = I$.) Similarly, our matrix E has strange rows 1, 2, 4, and 8, and the other rows form I_{11}:

$$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} & E_{17} & E_{18} & E_{19} & E_{1,10} & E_{1,11} \\ E_{21} & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} & E_{27} & E_{28} & E_{29} & E_{2,10} & E_{2,11} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_{41} & E_{42} & E_{43} & E_{44} & E_{45} & E_{46} & E_{47} & E_{48} & E_{49} & E_{4,10} & E_{4,11} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E_{81} & E_{82} & E_{83} & E_{84} & E_{85} & E_{86} & E_{87} & E_{88} & E_{89} & E_{8,10} & E_{8,11} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now the key property $CE = 0$ forces us to make the first row

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix},$$

the second row

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix},$$

the fourth row

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

and the eighth row

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

One can then check that the last of the four key properties — that each column of E have at least three 1s, so that changing any bit in the original message changes at least three bits in the encoded message — is also satisfied. So this is a legitimate Hamming code to correct one-bit errors or detect one- or two-bit errors.