Question 1 (5 marks): **(Cryptographic Hash Functions)**[1] Consider the following proposal for a hash function h. Let p be a large prime and let g be a generator mod p. Represent messages as sequences x= $x_1, x_2, \ldots x_n$ where the x_i are numbers mod p. We define the hash of such a message x by $h(x) = g^{x_1+x_2+\ldots+x_n} \mod p$.

1. (2 marks) Show that this hash function is good for use as a message digest for error correction purposes, in the sense that it has a high probability of detecting errors if messages x are transmitted in the form $(x, h(x))$, and a message (x, y) that is received is treated as correct if $h(x) = y$.

   In particular, for concreteness, suppose that the messages Alice will be sending are all of length two ($x = x_1, x_2$), and are generated uniformly at random.

   Suppose also that we know that communications channel that we are using contains occasional bursts of random noise (e.g., from sunspots on a radio channel) that may potentially damage every bit of the message $(x, h(x))$, including the hash. When Alice sends a message $(x, h(x))$, with

*Version 2 makes some minor typo fixes, see footnote.

[1](v2) The probability in this question revised to be called q rather than p, which clashed with the prime p.

probability q, the channel delivers the message to Bob exactly as transmitted. With probability $1-q$, the channel delivers to Bob a message (y, z) selected uniformly at random, where $y = y_1, y_2$ and y_1, y_2, z are all numbers mod p. (The channel never delivers a message to Bob if Alice did not send a message.) Bob accepts a message (y, z) that he receives as a correct transmission of a message y from Alice if $h(y) = z$, and treats it as corrupted otherwise.

If Bob receives a message and hash (y, z) and accepts it as correct, what is the probability that y is not the message sent by Alice?

2. (3 marks) Which of the three properties of cryptographic hash functions (pre-image resistant, second pre-image resistant, and collision-resistant[2]) are satisfied by the function h? Explain your answers. In case you say that the property is satisfied, give reasons to believe that it is. In case you say that the property is not satisfied, give a proof that that it is not satisfied (i.e., explain how an attacker could efficiently do what the property says cannot be done efficiently).

Question 2 (5 marks): **(Digital Signatures)** Suppose that h is a hash function taking long messages as input and producing 256 bit outputs and that M is a long message. Consider the following idea for constructing a signature scheme:

- A private signature key K_s is a pair of randomly generated sequences $x_1, ..., x_{256}$ and $y_1, ..., y_{256}$, both of length 256, where each value x_i and y_i is a 256 bit message.

- The corresponding public verification key K_v is the pair of sequences $h(x_1), ..., h(x_{256})$ and $h(y_1), ..., h(y_{256})$
- To sign message M, where the hash $h(M)$ is the sequence of bits $b_1...b_{256}$, define $sign(M, Ks) = (M, z_1...z_{256})$ where $z_i = x_i$ if $b_i = 0$ and $z_i = y_i$ if $b_i = 1$.

1. Explain how you could verify this signature is correct: what does the verification function $V(K_v, (M, r_1 \dots r_{256}))$ do to check that $(M, r_1 \dots r_{256})$ is message M signed using the signature key corresponding to K_v?
2. Explain why this scheme is secure - why is it hard for an attacker to forge a signed document that passes the test? Clarify what assumptions on the hash function are needed for your argument.

[2](v2) Was "collision-free" in version 1. In the literature, this expression is also used for the same concept, but suggests that there are no collisions rather than that they are hard to find, so collision-resistant is closer to the intended meaning.

3. Is it safe to use the same private key to sign more than one message? If not, explain what an attacker could do to attack a user who does this.

Question 3 (5 marks): **(Monetary Supply Laws)** One of the ways in which currencies (both standard and crypto-) differ is in their rules concerning how money is created. Different rules have different economic consequences and incentivize users and investors in potentially complex ways. One important economic metric is the *inflation rate*, which is the annual rate at which the cost of goods and services increases. As we have experienced in recent years, the factors impacting inflation are diverse and unpredictable, and may include pandemics, wars and catastrophic weather events. However, it is clear that one of the main reasons for the presently high inflation, and its consequences on people's lives, is the amount of new money that was created by central banks during the pandemic. Central bank doctrine in recent decades has been that deflation (decreasing cost of goods over time) is bad, because people tend to hoard money rather than spend, since they know they can buy the same goods cheaper later, and this causes unemployment as the economy grids to a halt. On the other hand, nobody likes high inflation when wages are fixed. Many central banks take the view that 2-3% inflation is socially optimal.

In this question, we focus on the *monetary inflation rate* r as a proxy[3] for price inflation. For a given period, this rate is defined as $r = (M' - M)/M$, where M is the amount of money in existence at the start of the period and M' is the amount of money in existence at the end of the period.

Cryptocurrencies use a number of rules to determine how much new money is created per block. In this question , we calculate the consequences for the monetary inflation rate. Let I_k be the amount of *new* money issued in the k-th block, so that I_0 is the amount of money issued in the genesis block of the cryptocurrency. Write M_k for the total amount of money that has been issued in the first k blocks. That is, $M_k = I_0 + \ldots + I_k$. Thus, the rate of inflation in the $k+1$-th block is $r_{k+1} = (M_{k+1} - M_k)/M_k$.

(a) **(A Bitcoin-like supply law, with halving)** Suppose that $I_0 = 2^N$, and

$$I_{k+1} = \begin{cases} I_k/2 & \text{if } I_k > 1 \\ 0 & \text{otherwise} \end{cases}$$

Write a closed form solution for M_k in terms of N, and use this to derive an expression for the rate of inflation r_{k+1} in the $k+1$-th block.

(b) **(An Ethereum1.0-like supply law)** Suppose $I_{k+1} = c$, where c is a constant. Write a closed form solution for M_k in terms of I_0, and use this to derive an expression for the rate of inflation r_{k+1} in the $k+1$-th block.

[3] In practice, population growth means that the cost of goods and services can be increasing due to increased competition for limited supply, even if the money supply is fixed.

(c) In case (b), what happens to r_k in the limit, as $k \to \infty$?

(d) **(A supply law that pays interest to savers/stakers)** Suppose that in each period, a fraction S of the total amount of money is saved, and new money is created by paying interest at a fixed rate R to the savers. (The interests is a reward for making available the saved money for some socially productive purpose. In "proof of stake" cryptocurrencies, the savers are the stakers, and the socially productive purpose is the work done by them to secure the currency.) That is $I_{k+1} = M_k SR$. Write a closed form solution for M_k in terms of I_0, and use this to derive an expression for the rate of inflation r_{k+1} in the $k+1$-th block.

(e) In case (d), what happens to r_k in the limit, as $k \to \infty$?

(Hint: one of the lecture slides shows an approach to finding a closed form, not involving a summation, for $1 + x + x^2 + \ldots + x^n$.)