Question 1

4 / 4 pts

Select each correct statement.

- [x] If a cryptographic hash function is collision resistant, then it is preimage resistant.
- [ ] If a cryptographic hash function is preimage resistant, then it is collision resistant.
- [ ] HMAC uses almost-universal hashing to produce an authentication tag.
- [ ] Wegman-Carter authentication uses a cryptographic hash to produce an authentication tag.
- [ ] Fast cryptographic hashes are faster than fast almost-universal hashes.

Question 2

4 / 4 pts

Recall that divisionless modular reduction computes the mod of 2^a-b without using division. What mod is being performed by the following code snippet? Give your answer by telling me the a and b of the modulus.

```
x = (x >> 8) + (x & 0xFF);
```

a = 8

b = 1

no multiply (multiply by 1)

x >> 8

```
divisionless_mod(acc, p = 2^a - b)
  hi = acc >> a
  lo = acc & 0b 1111...111   (a)
  return hi * b + lo   <= best if b easy
```

```
hi = x >> 8
lo = x & b1000
```

y * 1

$2^a - b$ a = 8 b = 1

Question 3

4 / 4 pts

Consider the following version of Horner's method which computes a polynomial with coefficients a_1, a_2, ..., a_n and variable k.

```
acc = 1
for i = 1 to n
   acc *= k
   acc += a[ i ]
return acc
```

Determine precisely what polynomial is being computed and answer the following questions about it.

What is the degree (ie, k's exponent) of the highest-degree term? n

What is the coefficient of the highest-degree term? 1

What is the degree (ie, k's exponent) of the lowest-degree term? 0

What is the coefficient of the lowest-degree term? a[n]

$$acc = 1$$

$i = 1$

$$acc = 1 * k$$

$$acc = (1 * k) + a[1]$$

$$= k + a[1]$$

$i = 2$

$$acc = (k + a[1])\,k$$

$$= k^2 + a[1]\,k$$

$$acc = (k^2 + a[1]\,k) + a[2]$$

$i = 3$

$$acc = (k^2 + a[1]\,k + a[2])\,k$$

$$= k^3 + a[1]\,k^2 + a[2]\,k$$

$$acc = k^3 + a[1]\,k^2 + a[2]\,k + a[3]$$

Question 4

4 / 4 pts

This problem will test your understanding of the sponge construction by having you simulate it. The internal function used will be the permutation p : $\{0,1\}^8 \rightarrow \{0,1\}^8$ where p(x) = x <<< 1 (an 8-bit permutation where x is rotated left one bit). We will use rate R = 4 bits and capacity C = 4 bits.

Let's say that after padding your data is 10101001. After the second invocation of p (ie, after absorbing this data), what is the value of your chaining block?

Give your answer as a sequence of bits without spaces or other characters (ie, use the characters 0 and 1 for your answers and nothing else).

10100011

$p: \{0,1\}^8 \rightarrow \{0,1\}^8 \qquad p(x) = x <<< 1$

R = 4 bits C = 4 bits

Data = 1 0 1 0 | 1 0 0 1