C - Cybersecurity C 2020-2021 - BEARSO - 10-24-2020

Questions 1 through 33 form the written portion of the cybersecurity event. It counts for 50% of the event score. The questions will be a mixture of Cryptography and Web Architecture.

Questions 34 and 35 form the hands-on cryptography questions. They count for 25% of the event score.

For the hands-on programming portion, go to https://www.hackerrank.com/bearso-2020-programming-hands-on (https://www.hackerrank.com/bearso-2020-programming-hands-on) and follow the instructions in this document: https://docs.google.com/document/d/1U5d6LedWSXWXi48io64GP_bopN5HOG3xugw2MgXlohs/edit?usp=sharing (https://docs.google.com/document/d/1U5d6LedWSXWXi48io64GP_bopN5HOG3xugw2MgXlohs/edit?usp=sharing). That counts for 25% of the event score.

1. (2.00 pts)
You've decided to encrypt a message with a substitution cipher. Instead of just scrambling the letters A-Z, you're going to scramble all the printable ASCII characters (of which there are 95), and your key is a random permutation. How many bits of security does your key have?

A) 26

B) 95

C) 125

D) 259

[✓ E) 491]

F) 9025

2. (1.00 pts) Which of the following are *hashing* algorithms? (You may select zero, one, or multiple.)

(Mark **ALL** correct answers)

[✓ A) SHA256]

☐ B) Electronic Codebook

☐ C) Vigenère

☑ D) MD5

☐ E) HTTPS

☑ F) Whirlpool

3. (6.00 pts) Consider the following encryption scheme:

Alice and Bob share a random secret password, **PASS**, that is 500 characters long. For Alice to sent a message **M** to Bob, she:

1. Divides **M** into chunks C_i of two characters each
2. Computes $H_i = SHA256(C_i + PASS)$ for each chunk, where + is string concatenation.
3. Alice sends these hashes to Bob, in order.

For Bob to read the message, he:

1. Loops through each hash, H_i, and:
2. tries each possible two-character string c until he finds $H_i = SHA256(c + PASS)$
3. His final message is the concatenation of all his c's.

Which of the following are problems with this scheme?

(Mark **ALL** correct answers)

☐ A) Bob is likely to end up with a different string that what Alice meant to encode.

☐ B) A 500 character password is not long enough for this type of scheme.

☐ C) SHA256 is one-way, which means Bob can't find the c's in any reasonable amount of time.

☑ D) This encryption has no diffusion, and patterns will show up in the output.

☑ E) The encrypted message has many more bits than the original, which would increase their bandwidth usage.

☐ F) **PASS** can't actually be random, because computers only use pseudorandom numbers, so attackers could guess it.

4. (6.00 pts) Which of the following would significantly improve the encryption scheme from the previous question?

(Mark **ALL** correct answers)

☐ A) Alice and Bob instead use the function $H_i = AES256(C_i, PASS)$.

☐ B) Alice randomly generates a new **PASS** for each message, to avoid key re-use.

☑ C) Alice and Bob instead use the function $H_i = SHA256(C_i + PASS + i)$.

☐ D) Alice and Bob instead use the function $H_i = SHA256(SHA256(C_i + PASS) + PASS)$.

☑ E) Alice and Bob instead use the function $H_i = SHA256(C_i + H_{i-1})$ for any $i > 2$.

☐ F) Using a secret **PASS** that is a multiple of 256 characters, so that it's more compatible with SHA256.

5. (2.00 pts) Which of the following are differences between the XOR operation $\oplus$, and the addition operation $+$ mod 256?

(Mark **ALL** correct answers)

☑ A) XOR is its own inverse (idempotent), but addition is not.

☐ B) Addition is associative, but XOR is not.

☐ C) XOR is invertible, addition is not.

☑ D) Addition has diffusion, XOR does not.

☐ E) Computers can do XOR much more quickly than addition.

☐ F) XOR is more pseudorandom than addition.

6. (3.00 pts) _____________ is the process of creating a new shared secret password, that eavesdroppers will not be able to figure out based on your conversation.

Key exchange

7. (2.00 pts) Which of the following best describes the use of a *certificate* in cryptography, in particular for websites?

- ○ A) A certificate proves that the content has been encrypted, by the agreed upon protocol. This stops downgrade attacks.
- ○ B) A certificate proves that the content is newly generated, with a verifiable timestamp. This stops replay attacks.
- ○ C) A certificate provides the necessary information for the client to decrypt the data. Without it, eavesdroppers could read the traffic.
- [✓ D)] A certificate is a way for the server to prove they are who they say they are.
- ○ E) A certificate proves that the company has been vetted to use good practice in security, by a third-party like Digicert or the US government.

8. (4.00 pts)

A _____________ attack is when an attacker has captured the unencrypted and encrypted versions of some messages, and now wants to figure out the secret key.

A _____________ attack is when the attacker has sent their *own* message to be unencrypted, gotten it back decrypted form, and now wants to figure out the secret key.

A _____________ attack is when the attacker steals secrets from the server using other information than the normal protocol -- such as timing, sound, or power usage.

(All answers should be two words)

Known plaintext | Chosen ciphertext | Side channel

9. (7.00 pts)

The RC4 algorithm generates an arbitrarily long sequence of bytes, after being initialized with a key. This sequence of bytes can then be used as a stream cipher. It makes use of an array, S, that at all times contains a permutation of the numbers 0 to 255.

```
S := RandomPermutation([0,1,2 ... 255], key)
i := 0
j := 0
while True:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
```

The RC4 algorithm is widely regarded as a secure and trustworthy algorithm.Suppose the algorithm was changed to:

```
S := RandomPermutation([0,1,2 ... 255], key)
i := 0
j := 0
while True:
    i := (i + 1) mod 256
    j := (j + i) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
```

The change has been highlighted in red. Explain either why this would still be a trustworthy stream cipher, or why it would be vulnerable. Support your answer in terms of diffusion or confusion.

Expected Answer: Key point: the variable "j" no longer depends on the values in S. This means that S will undergo a *fixed* permutation every so often (every 512 loops), that does not depend on its current state, so the cipher will repeat quickly. 5 points for saying that it will repeat. 1 point for saying there is no longer diffusion of S into j. 1 point for saying that S will be permuted the same way every 512 loops. Incorrect answers would include things like, "It's still secure", "S stops being a permutation", "j will grow too quickly", "the key isn't used"...

10. (2.00 pts)
The advent of post-quantum cryptography has made necessary the distinction between *information-theoretic security* and *computational security.* Which type of security is "more secure"?

[✓ A) Information-theoretic security]
B) Computational security

11. (2.00 pts) Of the two types of security listed in the previous question, one of them **cannot** be carried out with traditional internet communication. Which is that?

[✓ A) Information-theoretic security]
B) Computational security

12. (2.00 pts)
Give an example of how each of these two types of security were used in the year 2000 -- after modern cryptography, but before any quantum communications existed.

Expected Answer: Computational security: Anything involving hashing, RSA, public key, AES, etc -- most of the topics on this test. Information-theoretic security: One-time pads (and pretty much only those).

13. (9.00 pts)
Consider the following encrypt function, written in pseudocode. Identify three flaws with this system. Each flaw will get you three points. You can assume that this key is used only for this one message, i.e. you don't need to worry about key reuse.

NOTE: You will use this same code again in the Hands-On portion, with a different question. You will be given a copy of the code again. This might be relevant if you're deciding whether or not to attempt this problem.

```
import numpy
import random

#Get a random matrix, given a random number generator and a size.
#Fills it with random integers mod "modulo".
def Get_Random_Matrix(rand, modulo, size):
        mat = numpy.zeros((size,size))
        for i in range(size):
                for j in range(size):
                        mat[i,j] = numpy.floor(rand.random() * modulo)

        return mat

#Encrypt a message given the key.
#Requirement: "message" is an array of bytes, with length a multiple of 4.
def encrypt(message, key):

        #The number of rows and columns in the matrix
        n=4
        random.seed(key)
        output = []

        #Loop through every 4 bytes
        for block in range(0,len(message),n):

                matrix = Get_Random_Matrix(random, 256, n)
                vector = message[block : block+n]

                #This is matrix multiplication, mod 256
                vector = numpy.dot( matrix, vector ) % 256
                output += list(vector)

        return output
```

Expected Answer: Intended flaws: * The matrices might not always be invertible. In this case, you can't recover the plaintext. * The series of bytes [0,0,0,0] will always be mapped to [0,0,0,0] again. * The python random number generator is not secure. ...and certainly any other new flaws they find should count! Things that are not flaws: * The key isn't hashed. * Someone might use a small key. * It maps the same set of bytes to the same set, every time, like ECB (it doesn't) * Linear operations are easy to undo * n=4 is too small to be secure

14. (2.00 pts)
Diffie-Hellman is often carried out using exponentiation of integers, but this has some people concerned that it could be broken using advanced number theory. So instead, some people use Diffie-Hellman based on a collection of points, and "adding" those points together. What is this set of points called?

Elliptic Curve

15. (4.00 pts)
"Post-quantum" cryptography refers to cryptographic protocols developed to address the rise of quantum computers. However, very few people use these protocols yet. What are the main reasons for their lack of adoption?

(Mark **ALL** correct answers)

- ☐ A) Physicists don't understand the laws of quantum mechanics well enough to know if the algorithms work.
- ☑ B) The protocols are several times slower than traditional protocols like RSA.
- ☐ C) Most people don't have a quantum computer yet, so they can't even run the algorithms!
- ☑ D) These algorithms are new, so they haven't been studied for as long, so people don't trust them yet.
- ☑ E) It takes a lot of work to upgrade these protocols that are core to the internet, and get everyone to switch over.
- ☐ F) The US Military issues export restrictions on cryptographic algorithms, which mean you can't use the new protocols outside of the US.

16. (1.00 pts) Which numbers would you likely find in an RSA private key, but not in a public key?

(Mark **ALL** correct answers)

- [✓ A)] q
- [ ] B) n
- [ ] C) e
- [✓ D)] d
- [✓ E)] phi
- [ ] F) m

17. (1.00 pts) "Traditional" ciphers like substitution ciphers or Vigenere ciphers can have lots of keys, but are now considered a poor choice for secure communication. Why?

(Mark **ALL** correct answers)

- [✓ A)] They map the same input to the same output over and over, so you can easily find repeated letters or words.
- [ ] B) They were developed for English text, so there's no way to use them for other data like images or movies on a computer.
- [✓ C)] There's no confusion or diffusion in the encryption process, so you figure out the keys one part at a time.
- [ ] D) Computers can brute-force all of the keys and find the plaintext easily.

18. (1.00 pts) Which attack is most associated with session management and cookie abuse?

- A) Server-side Template Injection
- B) Buffer Overflow
- [✓ C)] Cross-site Request Forgery
- D) Cross-site Scripting

19. (2.00 pts) Identify all valid HTTP request verbs in the list below.

(Mark **ALL** correct answers)

- [✓ A)] PUT
- [ ] B) SEARCH
- [✓ C)] TRACE
- [✓ D)] OPTIONS
- [✓ E)] PATCH
- [ ] F) LISTEN

20. (1.00 pts) Internet standards, which describe how internet-related processes should operate are documented by

- [✓ A)] Request For Comments
- B) Transmission Control Protocol
- C) Internet Protocol
- D) Hypertext Transfer Protocol

21. (2.00 pts) An attacker is testing a website and tries the request

```
http://somewebsite.com/?file=../../../../../../etc/passwd
```

What vulnerability is the attacker looking for?

A) XML External Entities
[✓ B) Local File Inclusion]
C) Cross-site Scripting
D) SQL Injection

22. (2.00 pts) In the list below, identify both the character placed as a marker before a list of query parameters and the character that separates query parameters in a URL.

(Mark **ALL** correct answers)

A) #
[✓ B) &]
C) :
[✓ D) ?]
E) %
F) $

23. (2.00 pts) A cookie with the attribute ___________ is not accessible by JavaScript.

[✓ A) HttpOnly]
B) Secure
C) Max-Age
D) Domain

24. (1.00 pts) The ________ header of an HTTP request allows a server to identify information such as the application and operating system a request was made from.

User-Agent

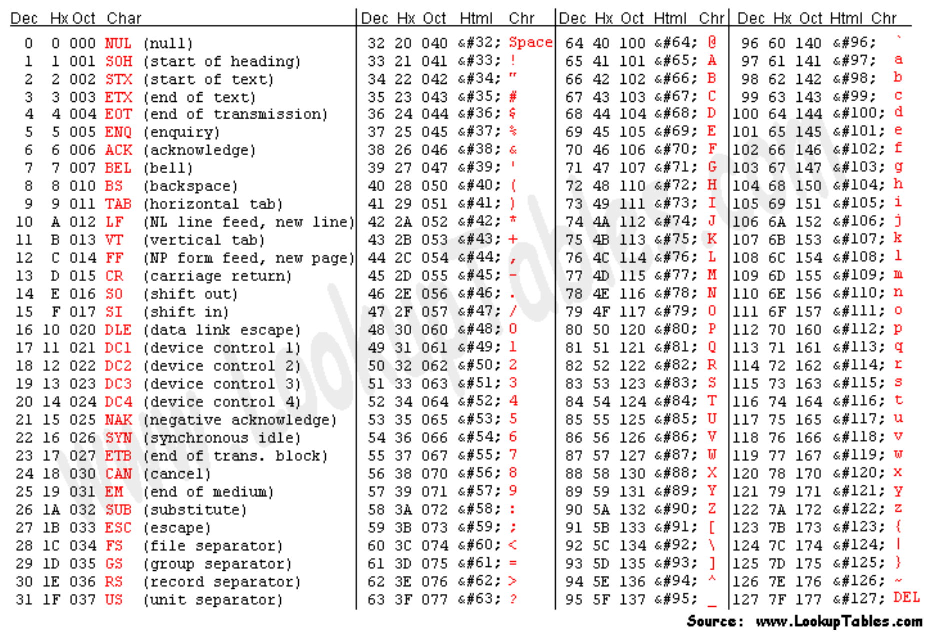
25. (1.00 pts)
CTF websites are infamous for overloading in the first hour of the event. In this case, you might encounter errors of the form __xx while desperately refreshing the page.

5

26. (2.00 pts) The URL-encoded version of the string "bearso{%signs=cool}" is _______________ .

Ignore the quotes when encoding and type your answer as a string without quotes around it.

Hint: This table may be useful.



Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH	(start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX	(start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX	(end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT	(end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ	(enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK	(acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL	(bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS	(backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB	(horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT	(vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR	(carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO	(shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI	(shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE	(data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1	(device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2	(device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3	(device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4	(device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK	(negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN	(synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB	(end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN	(cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM	(end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB	(substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC	(escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS	(file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS	(group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS	(record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US	(unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

bearso%7B%25signs%3Dc

27. (1.00 pts) What cookie attribute differentiates session cookies and persistent cookies? Enter either its exact name or a description of it.

Expected Answer: Exact name - Expires Description - expiration date or similar

28. (3.00 pts) What is the purpose of a webpage's robots.txt file?

Expected Answer: Sample answer: It specifies how to inform the web crawlers about which areas of the website should not be processed or scanned.

29. (3.00 pts) Explain the difference between reflected and persistent cross-site scripting attacks.

Expected Answer: Sample answer: Persistent XSS refers to unsanitized data stored in a server's database; anyone who accesses the resource can be affected. Reflected XSS is not stored in a database, but rather rendered immediately. Usually, a client sends in unsanitized data and it's displayed in the browser; it doesn't affect many people outside the user. Key points: reflected - attack payload not stored, affects a small amount of people (usually just the client), feedback is immediate persistent - attack payload stored, affects anyone who accesses the resource (many people), feedback occurs when affected resource is accessed

30. (3.00 pts) Consider this JavaScript function.

```
function f(s) {
  var res = "";
  for (i = 0; i < s.length; i++) {
    if (s[i] >= 'a' && s[i] <= 'z') {
      res += s[i].toUpperCase();
    } else if (s[i] >= 'A' && s[i] <= 'Z') {
      res += s.charCodeAt(i) - 'A'.charCodeAt();
    } else {
      res += s[i];
    }
  }
  return "bearso{" + res + "}";
}
```

What does `f("bDErF_FA_FcEry")` return?

bearso{B34R5_50_5C4RY}

31. (8.00 pts)
Consider an application with this snippet of code:

```
req = "SELECT * FROM users WHERE name='" + username + "' AND password='BEARSO'"
sqlserver.execute(req)  # This line of code sends the string req to an SQL server as a request.
```

a. (1 point) This app is vulnerable to a(n) __________________.

b. (3 points) To retrieve all elements of the table `users` what should the value of the variable `username` be?

c. (4 points) If an attacker wanted to steal data from another table, let's say `pocky-flavors`, what should the value of the variable `username` be? (assume the `pocky-flavors` table has the same dimensions as `users` as to not trigger an error).

Please add a., b., and c., before each answer in your response.

Expected Answer: a. SQL injection b In general: ‘ OR {true expression} {comment delimiter “;” or “--”} ex. ' OR 1=1; c. In general: ‘ UNION SELECT * FROM pocky-flavors WHERE {true expression} {comment delimiter “;” or “--”} ex. ‘ UNION SELECT * FROM pocky-flavors WHERE 1=1-- Comment delimiters are necessary since otherwise AND password='BEARSO' would be included in the query, which would result in nothing being returned.

32. (6.00 pts) Consider an application with this snippet of code:

```
if ($_POST["secret"] == $SUPERSECRET) {
    admin_login();
} else {
    get_mad();
}
```

a. (1 point) An attacker can exploit ____________________ in the conditional of the if statement to gain access to admin_login().

b. (2 points) A value of `secret` that an attacker could supply is _______________. (Assume `$SUPERSECRET` is a string.)

c. (3 points) Changing the program by one character to this program could fix its vulnerability. What should the programmer do?

Please add a., b., and c., before each answer in your response.

Expected Answer: a. Type juggling, ==, loose typing, or similar b. 0 c. Change the == in the conditional to === Please add a., b., and c., before each answer in your response.

33. (3.00 pts) This HTTP Request is missing a header. Determine which one it is, and fill out the line with both the header and its appropriate value.

```
POST /cookiejar.php HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.bearsobakery.com
Content-Type: application/x-www-form-urlencoded
__________________
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

order=chocolatechip
```

Content-Length: 19

Cryptography Hands-On Portion

34. (24.00 pts)
Consider the following encrypt function, written in Python. It is identical to the system you saw in the Written portion. All blocks are available for copy-paste at https://pastebin.com/raw/BsUzPmdX (https://pastebin.com/raw/BsUzPmdX) .

```
import numpy
import random

#Get a random matrix, given a random number generator and a size.
#Fills it with random integers mod "modulo".
def Get_Random_Matrix(rand, modulo, size):
        mat = numpy.zeros((size,size))
        for i in range(size):
                for j in range(size):
                        mat[i,j] = numpy.floor(rand.random() * modulo)

        return mat

#Encrypt a message given the key.
#Requirement: "message" is an array of bytes, with length a multiple of 4.
def encrypt(message, key):

        #The number of rows and columns in the matrix
        n=4
        random.seed(key)
        output = []

        #Loop through every 4 bytes
        for block in range(0,len(message),n):

                matrix = Get_Random_Matrix(random, 256, n)
                vector = message[block : block+n]

                #This is matrix multiplication, mod 256
                vector = numpy.dot( matrix, vector ) % 256
                output += list(vector)

        return output
```

Someone has used the *encrypt* function in the previous question to encrypt 8 bytes. Their secret key is the integer `28`. The output (the ciphertext) is the array `[5.0, 5.0, 43.0, 123.0, 68.0, 246.0, 201.0, 35.0]`. Implement a decrypt function, to figure out:What are the first two numbers of the input (the plaintext)? You may wish to use the following code, for computing matrix inverses modulo p.

```
from numpy import matrix
from numpy import linalg

def modMatInv(A,p):       # Finds the inverse of matrix A mod p
  n=len(A)
  A=matrix(A)
  adj=numpy.zeros(shape=(n,n))
  for i in range(0,n):
    for j in range(0,n):
      adj[i][j]=((-1)**(i+j)*int(round(linalg.det(minor(A,j,i)))))%p
  return (modInv(int(round(linalg.det(A))),p)*adj)%p

def modInv(a,p):          # Finds the inverse of a mod p, if it exists
  for i in range(1,p):
    if (i*a)%p==1:
      return i
  raise ValueError(str(a)+" has no inverse mod "+str(p))

def minor(A,i,j):    # Return matrix A with the ith row and jth column deleted
  A=numpy.array(A)
  minor=numpy.zeros(shape=(len(A)-1,len(A)-1))
  p=0
  for s in range(0,len(minor)):
    if p==i:
      p=p+1
    q=0
    for t in range(0,len(minor)):
      if q==j:
        q=q+1
      minor[s][t]=A[p][q]
      q=q+1
    p=p+1
  return minor
```

```
You may wish to use https://repl.it/languages/python (https://repl.it/languages/python) or https://repl.it/languages/python3 (https://repl.it/languages/py
thon3), they will have all the packages you need.
```

49

16

35. (24.00 pts)
We've been intercepting transmissions from aliens for a while, and they use a special language with only ten characters: ACEFLMOPXY. By taking large samples, we have their digraph frequencies -- the frequency of every pair of letters when in this language. The matrix of frequencies is:

```
[[2.71743254e-03, 2.95467253e-04, 1.40289673e-03, 7.20653007e-06,
        1.85226513e-02, 3.79145035e-03, 3.03941946e-04, 1.31030842e-03,
        2.08116563e-02, 4.08118653e-03],
       [5.86397974e-03, 5.00155945e-03, 1.76121250e-02, 2.49812154e-02,
        1.01608926e-03, 6.77517525e-04, 2.10792531e-02, 2.12760389e-03,
        6.94094301e-04, 2.74646235e-02],
       [1.30648789e-03, 4.30724718e-04, 1.81207876e-03, 4.00819280e-05,
        7.62017704e-03, 3.40079355e-03, 1.97508670e-02, 1.93630388e-02,
        2.13334334e-02, 8.90379568e-03],
       [6.03752785e-03, 6.79558987e-04, 5.78578167e-04, 1.89196884e-02,
        3.90869323e-02, 1.34144667e-06, 5.46029978e-05, 3.43034643e-02,
        1.57870130e-03, 4.48519966e-07],
       [1.08794700e-02, 5.48098533e-02, 6.69358702e-03, 1.74655811e-05,
        1.69510857e-06, 2.04915851e-02, 2.01802310e-04, 4.80953473e-02,
        5.05292049e-03, 1.95240540e-02],
       [6.32580292e-03, 2.71218352e-02, 2.67347435e-02, 3.00882655e-05,
        2.87534777e-04, 4.66718544e-05, 1.45967595e-02, 6.16627671e-04,
        5.45688335e-05, 1.27962918e-03],
       [1.33773698e-03, 1.47335003e-03, 1.02548815e-02, 3.98417829e-03,
        1.46758075e-02, 5.70272782e-04, 1.62947773e-03, 2.90758251e-02,
        8.14832182e-04, 1.91533197e-02],
       [1.86562469e-02, 3.37085700e-03, 9.25444336e-04, 3.81832477e-02,
        5.12699314e-02, 8.12671061e-03, 5.91072348e-04, 8.49518905e-04,
        1.74208748e-05, 1.37923656e-02],
       [2.84182925e-05, 1.14994712e-02, 5.69087979e-04, 5.01564480e-03,
        1.53687521e-04, 1.85763624e-02, 1.23625437e-02, 1.18727534e-05,
        1.13290886e-02, 1.64661646e-02],
       [9.10948547e-05, 1.83538397e-03, 1.73780557e-02, 1.00620275e-02,
        3.31332740e-02, 2.14115561e-02, 1.23993612e-02, 2.92084602e-05,
        1.43256257e-02, 6.74294919e-03]]
```

You can also view these numbers at https://pastebin.com/raw/BsUzPmdX (https://pastebin.com/raw/0VnPNbcc) . This could be useful if you need to copy-paste the text and cannot copy-paste from the browser.

Recently they started using a substitution cipher. We need you to match the character frequencies to decode this block of text. It's alien, so don't expect it to make much sense in English...

```
ofpmaafcfcpoxyaofmfpoffmlcyaaaelxxxclycyaomelpofpoelelclclxlcooxlcoxxlcoofxlyclcfcpefcyaaaomefxcpoxlxlxcoemlcfcofpomfpelfaelclfpmmelxxcpoelcyaecelcyafefex
lfafemfmafclpmaefaaefecomaeclpofafpelfelcomecoefpmefclclxlyafcfclcfcoelyaofclxcyaapefmafmpelemlyfmlyaooefcoxlcpefclcoxcpoxlxyafecfaemcmaexcaoxlclfapoecoel
pmelcfclxlxcfpmclclxxcyapoxlxcaoxlfpmefcfcaefmapofpmyclcfpmefcoxcpoefcomemlclclpoxxlclyapypoeclmafcomlclcoxoecpeycpoyaelcoefcpypmfcffmefapmlpmemclxyaoxlcl
comclcyycpofafcyypmeloecofapmcfmafxlfclyafpoxlcmelcaefpomlxlxlyaofpmcaaopofcoelxlfmemfcycoxxcoelclxlcpcmelcomffcycpoeclfpomapmcoxcpolxlcpmefpoefmlcyyapoya
ofmaaafxcoyclyaxcoeclypycoelfaapelcoxyfaaelyaofclfpoeclclxlpycoemlxlclfcpoemelfclfpelymfxxxlxlxlcfeffcoxcpoelxcmeclelfcpocoefclyapmaexyapmemlxcolcpefelcly
apoyapmfapofcllxlcomfmeapmexlffxxmafefclxlcpelxcpofpefpmfpooxaoelxxcmlcmcoelcoemecoxcoxyapyyapefclxlxclxlxxlcomecfpmeclxxcfpmafpoemeclcpmaexcoefeyafcfcyaf
elyaelclclxlclxcapoelfmfpexxlxlpoxlxxxxcfxcoxcoxxclycfafaapefapomafpypmcomlp
```

Once you've decrypted it, please encrypt the words "EPOXY FLAME FLEECE" and submit these as your answers.

mleaf

xcypm

xcmmom

You're done with the test!