



**Titans Invitational Coding Championship
2024
Questions**

St Theresa Catholic Secondary School

1) Password standards verification

Back in 2004, the “NIST Special Publication 800-63. Appendix A” created the password standard that annoys everyone who ever has to create a password. Your job is to write such an annoying password verifier. It will look at each password candidate and judge them as ‘valid’ if they comply with the following specifications and ‘not valid’ while listing the *first* specification that they have not met from the following list.

Some invalid passwords will fail to comply with multiple specifications. Only state the first specification that it misses (according to the order below).

The specifications are that password must:

- 1) be at least 8 characters in length
- 2) contain at least one lowercase letter
- 3) contain at least one uppercase letter
- 4) contain at least one numerical digit
- 5) contain at least one of the following special characters ! @ # \$ % ^ & * < > ?
- 6) not contain alphabetically consecutive letters
- 7) not contain numerically consecutive digits
- 8) not contain consecutive repeated letters or digits

Sample input	Sample output
1234	not valid, password must be at least 8 characters in length
BadApple	not valid, password must contain at least one numerical digit
M8b&zR9k	valid
*8WaTu889	not valid, password must not contain alphabetically consecutive letters
*8WacT868	valid

Note that letters are considered alphabetically consecutive and digits are considered numerically consecutive if they are found in ascending or descending order. (45, cb, 32 and Xy are all considered alphabetically or numerically consecutive).

Hint→ if you find most of these specifications difficult, your program can achieve marks on some of the test cases if it accurately detects failure of some of the first conditions, for example, just check and report if it is less than 8 characters in length.

Remember, you must state the reason that a password is considered invalid.

There will be 10 passwords in each set of test data.

Test data Trial 1 is named DATA11.txt

Test data Trial 2 is named DATA12.txt

2) Zipper Cypher

A simple Zipper Cypher involves taking a plain text message, splitting it in half (after adding a blank space if needed to make it of even length) and then alternately inserting the characters into a new sentence.

For Example:

This is that

The diagram shows the plain text 'This is that' being split into two parts: 'This i' and 's that'. Arrows indicate the interleaving process: 'T' from 'This i' and 's' from 's that' form 'Ts'; 'h' from 'This i' and ' ' from 's that' form 'h '; 'i' from 'This i' and 't' from 's that' form 'it'; ' ' from 'This i' and 'h' from 's that' form ' sh'; 'a' from 'This i' and 'a' from 's that' form 'ai'; and 't' from 'This i' and 't' from 's that' form 't'. The final cypher text is 'Tsh itsh ait'.

Tsh itsh ait

In the test data, some of the cases are plain text to be zipped. Others are cypher text to be unzipped. Cyphertext cases will have a * at the beginning.

Sample Input	Sample Output
Look before you leap	Leo oyko ub elfeoarp
*Sheen dc lionw nts	Send in the clowns

Write a program that will read 10 lines of test cases from the test data and output either plain text or cypher text. You will know if it is cypher text if it starts with a *. Your program should remove the * before decoding.

Test data Trial 1 is named DATA21.txt

Test data Trial 2 is named DATA22.txt

3) Simple shift cypher

A simple shift cypher shifts every letter over by the same amount looping around at the ends of the alphabet. For example, shifted 2 letters to the right 'zoo' becomes 'bqq' and the sentence "i do like the zoo" becomes "k fq nkmg vjg bqq".

For this problem, there are 10 test cases, each of which are cyphered by shifting either left or right by any number from 1 to 25 inclusive. The number and direction that you shift by is constant in each case, but not constant from case to case. Each test data sentence is guaranteed to contain the word 'the' (unlike the second example below)

Sample input aol yhpu pu zwhpu mhssz thpusf pu aol wshpuz pdnaa xhejz ieya oaa dks pdau nqj	Sample Output the rain in spain falls mainly in the plains three blind mice see how they run
---	--

Write a program that will take 10 lines of cypher text and print it out in plain text.

Remember that the rule loops around at the end of the alphabet so if 'z' is shifted two to the right it becomes 'b'. Each line contains the word 'the'

Test data Trial 1 is named DATA31.txt

Test data Trial 2 is named DATA32.txt

4) Polynomial secret share

Adi Shamir invented Shamir's Secret Sharing (SSS), a method of sharing a secret amongst several agents. In his scheme, a predetermined number of agents must combine their 'shares' to determine the secret. Whenever a smaller group, less than the 'threshold', minimum number of agents, combine their shares, the secret is mathematically impossible to determine. Shamir's system uses finite field arithmetic. For simplicity we will use an easier version (also less secure) based on integer arithmetic.

SSS works like this. If the threshold is n , (that is, n agents must share to reveal the secret), a polynomial of degree $n-1$ is generated with random coefficients and a constant that is the secret. Each agent then receives a point on the polynomial. If n agents combine their points, they can determine the equation (and therefore the constant) of the polynomial.

For Example:

Secret :2125

Threshold:3

$a_0=2125$

$a_1= 5$ (chosen at random)

$a_2=7$ (chosen at random)

$p(x)=7x^2+5x+2125$

shares to distribute to agents:

(1,2137) (2,2163) (3,2203) (4,2257)

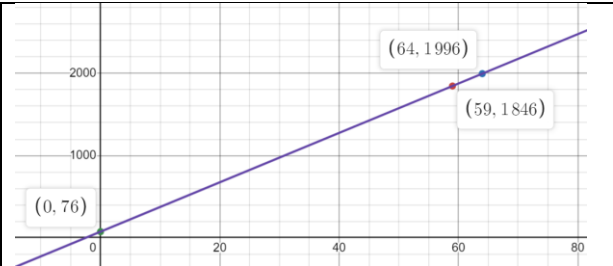
Now any 3 agents can combine their shares to determine the polynomial (and therefore the 'secret')

If the agent with (2,2163) works with the agents with (3,2203) and (4,2257):

Points (2,2163),(3,2203) and (4,2257) \rightarrow solve to get $p(x)=7x^2+5x+2125 \rightarrow p(0)=2125 \rightarrow$ the secret is 2125

The threshold can be anything from 2 to 4. (For partial points, you may want to just solve when the threshold is 2, these will be linear equations)

Each test case will contain the threshold number n on one line followed by n shares on the next n lines. The x and y coordinates are separated by a single space.

Sample input	Sample Output	
3	2125	
2 2163	76	
3 2203		
4 2257		
2		
59 1846		
64 1996		

Write a program that will determine the secret number for each of the 10 test cases.

Test data Trial 1 is named DATA41.txt

Test data Trial 2 is named DATA42.txt

5) Book Cypher DNA

A book cypher encodes plain text as a series of numbers indicating where to find the words or letters of the text within a book. For example:

The plain text 'meet me Wednesday' could be sent as 234 848 1023 if the known book had 'meet' as its 234th word, 'me' as its 848th word and 'Wednesday' as its 1023rd word.

The sender and receiver would have to both have full knowledge of the book being used to encode and decode the message.

Imagine an agent with full knowledge of another agent's DNA sequence. He could send a message as a series of numbers to be decoded using the recipient's DNA.

DNA strands contain the letters A, T, G, C (adenine, thymine, guanine, and cytosine).

If one knows that your DNA sequence is

GCCCGAGATTGTTGCTGATGAGTTGATGGCTGGTTCAGGTTGATGTCGACGAAGTCTAGAGTTGATAAAG

And chooses A=0 and T=1, they can encode a message using your DNA as a decoder key. If each number represents how far to move from the previous letter in the DNA sequence, starting at the beginning and looping around continuously:

Letting 0 represent a blank space and the numbers 1 to 26 represent the alphabet a-z and encoding every number as 5-bit binary, they could encode the message 'Sunday' as:

→ Sunday → 19 21 14 4 1 25

→ 10011 10101 01110 00100 00001 11001

→ TAATT TATAT ATTTA AATAA AAAAT TTAAT

→ 8, 9, 3, 2, 1, 3, 10, 3, 4, 1, 5, 6, 2, 5, 3, 2, 1, 11, 9, 3, 5, 11, 7, 6, 6, 2, 5, 3, 2, 12

As moving 8 letters from the beginning of the DNA strand gives you a T, moving 9 letters from the first T gives you A and so on. The 11 comes from moving 11 spaces including circling back to the beginning to get from the second last A in the strand to the next T.

GCCCGAGATTGTTGCTGATGAGTTGATGGCTGGTTCAGGTTGATGTCGACGAAGTCTAGAGTTGATAAAG

Each dataset contains a strand of DNA followed by 10 test cases. Each test case is either a word to be encoded or a numeric sequence to be decoded. The numbers of each numeric sequence are separated by commas. Your output must convert numeric sequences into the English words and English words into numeric sequences using the given DNA strand of each dataset and the scheme given in this question.

Sample input GCCCGAGATTGTTGCTGATGAGTTG ATGGCTGGTTCAGGTTGTATGTCGAC GAAGTCTAGAGTTGATAAAG sunday monday 1, 8, 1, 2, 5, 1, 10, 3, 4, 1, 5, 3, 3, 3, 4, 1, 2, 2, 12, 1, 8, 3, 2, 3, 11, 7, 6, 3, 1, 2, 2, 5, 3, 2, 12 1, 8, 1, 4, 1, 2, 11, 3, 4, 1, 5, 3, 3, 3, 2, 5, 1, 13, 1, 8, 3, 5, 1, 10, 3, 1, 3, 6, 6, 2, 1, 2, 2, 3, 2, 1, 1, 7, 2, 1, 1, 2, 6, 3, 2	Sample Output 8, 9, 3, 2, 1, 3, 10, 3, 4, 1, 5, 6, 2, 5, 3, 2, 1, 11, 9, 3, 5, 11, 7, 6, 6, 2, 5, 3, 2, 12 9, 1, 4, 3, 1, 10, 3, 1, 2, 2, 5, 6, 2, 5, 3, 2, 1, 11, 9, 3, 5, 11, 7, 6, 6, 2, 5, 3, 2, 12 tuesday Wednesday
---	---

Write a program that reads the dna sequence from the first line. And then the 10 cases. Each if the test case is an English word, convert it to a list of numbers as indicated above. If it is a list of numbers, determine the word that it represents. Use the dna strand from the first line as the decoder key for each test case in that trial.

Test data Trial 1 is named DATA51.txt

Test data Trial 2 is named DATA52.txt