



Instructions

You should write a program for part (a) of each question, and produce written answers to the remaining parts. The written questions can be solved without writing the program for part (a), although you may write a program to help produce these answers if you wish.

You may use a calculator and the on-line help that your programming language provides, and you should have a pen, some blank paper, and a blank floppy disk on which to save your programs. You must not use any other material such as disks, files on a computer network, books or other written information.

Mark the first page used for your written answers with your name, age in years and school/college. Number all pages in order if you use more than one sheet. All of your computer programs should display your name and school/college when they are run, and the floppy disk you use to submit the programs should also show your name and school/college.

For your programs to be marked, the source code must be saved, along with executables if your language includes a compiler. This includes programs used to help answer written questions. You must clearly indicate the name given to each program on your answer sheet(s).

Sample runs are given for parts 1(a), 2(a) and 3(a). Bold text indicates output from the program, and normal text shows data that has been entered. The output format of your programs should follow the 'sample run' examples. Your programs should take less than two minutes of processing time for each test.

Attempt as many questions as you can. Marks allocated to each part are shown in square brackets next to the questions. Partial solutions (such as programs that only satisfy some of the specified requirements, or partly completed written answers) may get partial marks. Questions can be answered in any order.

**Question 1** Some computer programs try to prevent easily guessable passwords being used, by rejecting those that do not meet some simple rules. One simple rule is to reject passwords which contain any sequence of letters immediately followed by the same sequence.

**Passwords**

For example:

APPLE	Rejected	(P is repeated)
CUCUMBER	Rejected	(CU is repeated)
ONION	Accepted	(ON is not immediately repeated)
APRICOT	Accepted	

**1 (a)** [22 marks] Write a program which inputs a password (between 1 and 10 upper case letters) and then prints **Rejected** if the password is rejected, or **Accepted** otherwise. Your program should then terminate.

*Sample run*

```

Password: BANANA
Rejected

```

**1 (b)** [5 marks] Suppose the only valid letters are A and B; the password ABA cannot be made longer since both ABAA and ABAB will be rejected. What is the length of the shortest password that cannot be made longer, if A, B and C are valid? Give an example.



**Question 3**  
*Playing games*

Romulus and Remus are playing a game where they take it in turns to throw a pair of dice and then add the two numbers. In order to make the game fair, the possible scores from Romulus' pair of dice and Remus' dice, and the number of different ways each of them can be formed, must be the same. Every side of a die consists of a single positive integer. [NB: This excludes 0.]

*note: 'die' is the singular of 'dice'*

For example, Romulus has a normal pair of six sided dice (each numbered {1,2,3,4,5,6}) and Remus has an unusual pair of dice: the first numbered {1,2,2,3,3,4}, and the second numbered {1,3,4,5,6,8}. This is fair because both pairs of dice can make the numbers from 2 to 12, and (for example) 6 can be made in 5 ways with each pair.

	1	2	3	4	5	6
1	2	3	4	5	<b>6</b>	7
2	3	4	5	<b>6</b>	7	8
3	4	5	<b>6</b>	7	8	9
4	5	<b>6</b>	7	8	9	10
5	<b>6</b>	7	8	9	10	11
6	7	8	9	10	11	12

	1	2	2	3	3	4
1	2	3	3	4	4	5
3	4	5	5	<b>6</b>	<b>6</b>	7
4	5	<b>6</b>	<b>6</b>	7	7	8
5	<b>6</b>	7	7	8	8	9
6	7	8	8	9	9	10
8	9	10	10	11	11	12

*Sample run*

```
6
2 4 1 3 5 6
6 4 2 3 1 5

1 2 2 3 3 4
8 6 5 4 3 1
```

**3 (a)**  
**[20 marks]**

Write a program that inputs the values on one of the pairs of dice and then outputs a different pair of dice that could be used to play a fair game. Two dice are only different if they have different sides; so {1,2,3,4,5,6} and {6,5,4,3,2,1} are not different. Similarly, a different order does not distinguish two pairs of dice; so {1,2,3} and {4,5,6} is not different to {4,5,6} and {1,2,3}.

The first line of input will consist of a single integer 'n' (1 <= n <= 8) denoting the number of sides all the dice will have. The second line will consist of 'n' integers (in the range 1 to 8) giving the values on the sides of the first die. The third line will also consist of 'n' integers (in the range 1 to 8) giving the values for the second die.

Your output should consist of an example solution if you believe there to be one (with one die given on each line), or the word **impossible** if you do not. Note that the example solution does not need to be sorted, and the value on a side can be any positive integer (i.e. values may be greater than 8).

**3 (b)**  
**[4 marks]**

Three equivalent pairs of dice have been mixed together. The six dice are labelled {1,5,6,7,9,10}, {2,3,4,6,7,8}, {2,6,7,8,10,11}, {1,2,3,5,6,7}, {3,7,8,9,11,12} and {3,4,5,7,8,9}. What were the original pairings?

**3 (c)**  
**[4 marks]**

Given two 'n' sided dice, what is the largest number of distinct sums that can be formed. What is the smallest?

**3 (d)**  
**[5 marks]**

Romulus and Remus have just found three three-sided dice whose sides use the numbers 1–9 (once each). If dice A and B are thrown, of the nine possible combinations that can come up, die A shows a higher value than die B for more than half the combinations. Surprisingly, if dice B and C are thrown, die B shows a higher value for more than half of the possible combinations *and* if dice A and C are thrown, die C shows a higher value for more than half of the possible combinations. If one of die A's sides is numbered 1, give a set of dice that satisfy these conditions.

**Total marks: 100.**