Accredited

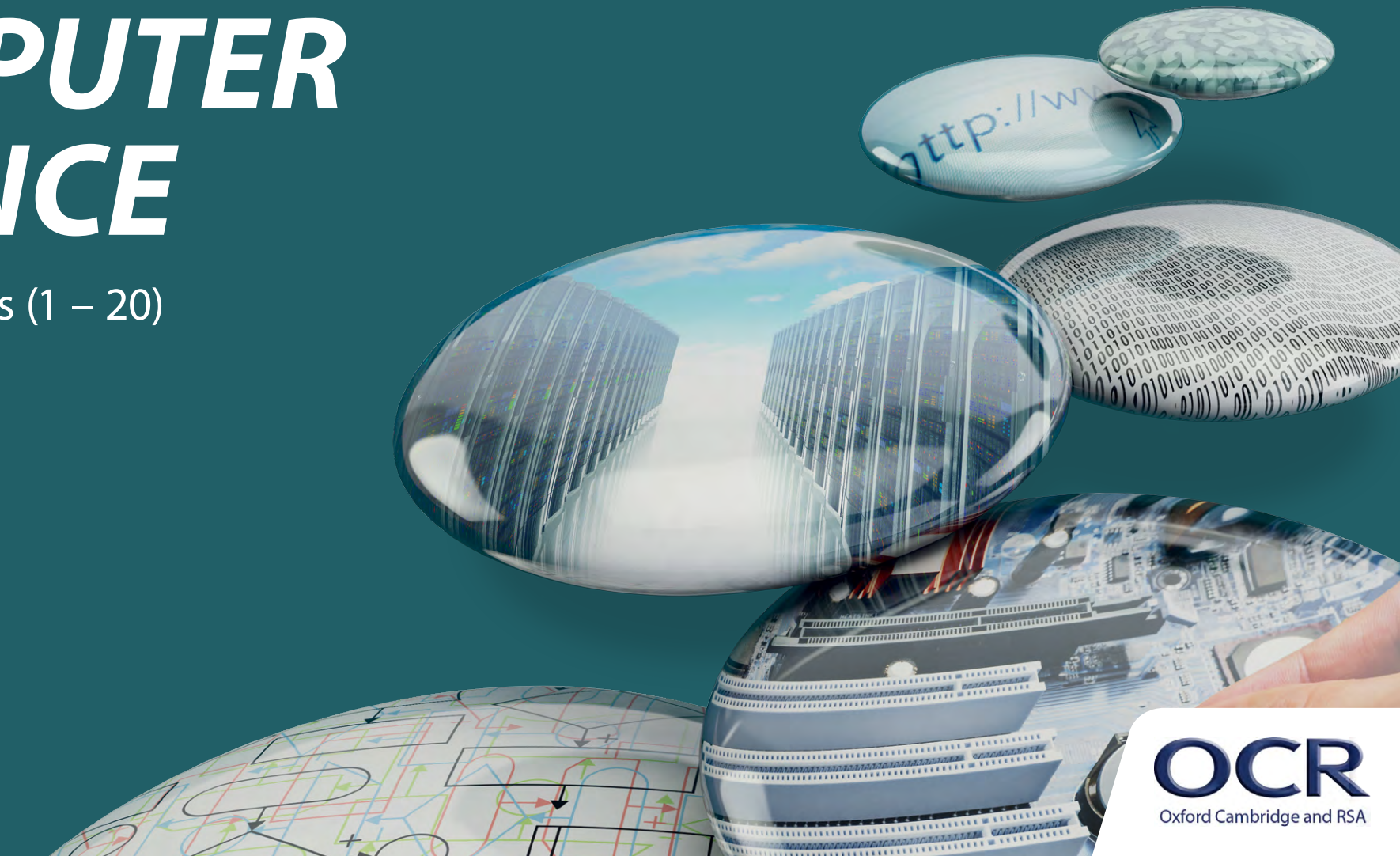
A LEVEL

H446

COMPUTER SCIENCE

Code Challenges (1 – 20)

August 2015



OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright © 2015 OCR. All rights reserved.

Copyright
OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee.
Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

CONTENTS

Factorial Finder	Page 4
Speed Tracker	Page 4
Thief!	Page 4
Classification	Page 4
Fruit Machine	Page 5
Unit Converter (temperature, currency, volume)	Page 5
Credit Card Validator	Page 5
Arithmetic test	Page 5
Happy Numbers	Page 5
Number Names	Page 6
Regex Query Tool	Page 6
Quiz Maker	Page 6
Caesar Cipher	Page 6
Events calendar	Page 6
Pangrams	Page 6
Kaprekar	Page 6
Number Table	Page 7
Years in a Range	Page 7
Logic Gate	Page 7
Palindromes	Page 7

1	**Factorial Finder** The Factorial of a positive integer, n, is defined as the product of the sequence n, n-1, n-2, ...1 and the factorial of zero, 0, is defined as being 1. Solve this using both loops and recursion.
2	**Speed Tracker** Create a program that takes a time for a car going past a speed camera, the time going past the next one and the distance between them to calculate the average speed for the car in mph. The cameras are one mile apart. **Extensions:** 1. Speed cameras know the timings of each car going past, through number plate recognition. Valid number plates are two letters, two numbers and three letters afterwards, for example XX77 787. Produce a part of the program that checks whether a number plate matches the given pattern. Tell the user either way. 2. Create a program for creating a file of details for vehicles exceeding the speed limit set for a section of road. You will need to create a suitable file with test data, including randomised number plates and times. You will then use the code you've already written to process this list to determine who is breaking the speed limit (70mph) and who has invalid number plates.
3	**Thief!** A thief has managed to find out the four digits for an online PIN code, but doesn't know the correct sequence needed to hack into the account. Design and write a program that displays all the possible combinations for any four numerical digits entered by the user. The program should avoid displaying the same combination more than once. Submit a fully detailed Showcase for your program.
4	**Classification** A simple classification system asks a series of Yes/No questions in order to work out what type of animal is being looked at. Eg Does it have 4 legs? Does it eat meat? Does it have stripes? These systems can often be drawn using a "tree" structure. Carry out some simple research on classification trees, then write a program to help the user decide between the following: horse, cow, sheep, pig, dog, cat, lion, tiger, whale, dolphin, seal, penguin, ostrich, sparrow, spider, ant, bee, wasp, termite, octopus, squid Is there a better way to do this than using 101 IF...ELSE...END IFs? Develop your classification system for your own area of interest: pop bands; pokemon; cars; footballers; teachers; diseases etc.

5	**Fruit Machine** Write a program to simulate a Fruit Machine that displays three symbols at random from Cherry, Bell, Lemon, Orange, Star, Skull. The player starts with £1 credit, with each go costing 20p. If the Fruit Machine "rolls" two of the same symbol, the user wins 50p. The player wins £1 for three of the same and £5 for 3 Bells. The player loses £1 if two skulls are rolled and all of his/her money if three skulls are rolled. The player can choose to quit with the winnings after each roll or keep playing until there is no money left.
6	**Unit Converter (temp, currency, volume)** Converts various units between one another. The user enters the type of unit being entered, the type of unit they want to convert to and then the value. The program will then make the conversion.
7	**Credit Card Validator** Takes in a credit card number from a common credit card vendor (Visa, MasterCard, American Express, Discoverer) and validates it to make sure that it is a valid number (look into how credit cards use a checksum).
8	**Arithmetic test** A primary school teacher wants a computer program to test the basic arithmetic skills of her students. Generate random questions (2 numbers only) consisting of addition, subtraction, multiplication and division. The system should ask the student's name and then ask ten questions. The program should feed back if the answers are correct or not, and then generate a final score at the end. **Extensions:** 1. Extend your program so that it stores the results somewhere. The teacher has three classes, so you need to enable the program to distinguish between them. 2. The teacher wants to be able to log student performance in these tests. The teacher would like the program to store the last three scores for each student and to be able to output the results in alphabetical order with the student's highest score first out of the three.
9	**Happy Numbers** A happy number is defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers. Display an example of your output here. Find the first eight happy numbers.

10	**Number Names** Show how to spell out a number in English. You can use a pre-existing implementation or make your own, but you should support inputs up to at least one million (or the maximum value of your language's default bounded integer type, if that's less). **Extensions:** 1. Create support for inputs other than positive integers (like zero, negative integers, and floating-point numbers).
11	**Regex Query Tool** This is a tool that allows the user to enter a text string and then in a separate text box enter a regex pattern. It will run the regular expression against the string and return any matches or flag errors in the regular expression.
12	**Quiz Maker** Make an application which takes various questions from a file, picked randomly, and puts together a quiz for students. Each quiz can be different and then reads a key to grade the quizzes.
13	**Caesar Cipher** Implement a Caesar cipher, both encoding and decoding. The key is an integer from 1 to 25. This cipher rotates the letters of the alphabet (A to Z). The encoding replaces each letter with the 1st to 25th next letter in the alphabet (wrapping Z to A). So key 2 encrypts "HI" to "JK", but key 20 encrypts "HI" to "BC".
14	**Events calendar** Create a menu driven program that allows the user to add or delete events from a list of dates and timings, just like a calendar. The program should warn you if any of the events overlap when entering them. **Extensions:** 1. Make it so that none of the events are hard-coded into the program
15	**Pangrams** **"The quick brown fox jumps over the lazy dog";** note how all 26 English-language letters are used in the sentence. Your goal is to implement a program that takes a series of strings (one per line) and prints either True (the given string is a pangram), or False if it is not.
16	**Kaprekar** Determine whether a number is a Kaprekar number or not. See http://mathworld.wolfram.com/KaprekarNumber.html for more information.

17	**Number Table** Write a program that takes a symbol (+,-,* or /) and a natural number (>0) and makes a table like below for the operation from 0 to n For this example the user has entered "+ 4": + | 0 1 2 3 4 ---------- 0 | 0 1 2 3 4 1 | 1 2 3 4 5 2 | 2 3 4 5 6 3 | 3 4 5 6 7 4 | 4 5 6 7 8
18	**Years in a Range** Write a program to count the number years in a range that has a repeated digit. For example, 2012 has a repeated digit, but 2013 does not.
19	**Logic Gate** Write a program that will give the students the answer to logic gate questions For example: Enter logic gate : OR Enter first input : 1 Enter second input : 0 Result = 1 It should work for the logic gates OR, AND, XOR, NAND and NOR
20	**Palindromes** Write a program that checks if a string entered by the user is a palindrome. A palindrome is a word that reads the same forwards as backwards like "racecar"

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

OCR Resources: *the small print*

OCR's resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

© OCR 2015 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content:
Thumbs up: alex_white/Shutterstock.com, Thumbs down: alex_white/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: resources.feedback@ocr.org.uk

OCR customer contact centre
General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored.

©OCR 2015 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England.
Registered office 1 Hills Road, Cambridge CB1 2EU. Registered company number 3484466. OCR is an exempt charity.