PROVISIONAL

20 CODE CHALLENGES

February 2015

GCSE (9–1)
Computer Science

GCSE REFORM

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.

This resource is an exemplar of the types of materials that will be provided to assist in the teaching of the new qualifications being developed for first teaching in 2016. It can be used to teach existing qualifications but may be updated in the future to reflect changes in the new qualifications. Please check the OCR website for updates and additional resources being released. We would welcome your feedback so please get in touch.

CONTENTS

1	**Mastermind** Generate a random four digit number. The player has to keep inputting four digit numbers until they guess the randomly generated number. After each unsuccessful try it should say how many numbers they got correct, but not which position they got right. At the end of the game should congratulate the user and say how many tries it took. **Extensions:** 1. Let the user pick an easy mode which shows the user which position that they guessed correctly 2. Let the user pick a hard mode that gives five digits instead of four 3. After the game is finished, ask the user for their name and input their score into a table. Show them the high score at the start of the game so that it gives a sense of competition.
2	**Averages** Make a program that asks the user for a series of numbers until they either want to output the average or quit the program. **Extensions:** 1. Expand the program to print the median and mode averages also 2. Include options so that if the user wants to, they can save their list of numbers to a text file and read them back out later on.
3	**Email validator** Make a program to check whether an email address is valid or not. For instance, you could make sure that there are no spaces, that there is an @ symbol and a dot somewhere after it. Also check the length of the parts at the start, and that the end parts of the address are not blank. **Extensions:** 1. When an email address is found to be invalid, tell the user exactly what they did wrong with their email address rather than just saying it is invalid 2. Allow the user to choose to give a text file with a list of email addresses and have it process them all automatically.

4	**Password reset program** Only accept a new password if it is: 1. At least eight characters long 2. Has lower case and upper case letters. The password reset program should also make the user input their new password twice so that the computer knows that the user has not made any mistakes when typing their new password. **Extensions:** 1. Make some sort of algorithm to suggest how strong the password is (Weak, Medium, Strong) depending on length, whether or not the password has special characters in etc 2. Let the user input their username. The program should go to a text file with a list of usernames and old passwords, and the program should only let you change your password if you input your old password.
5	**Basic lists** Make a program that lets a user input a series of names into a list. The program should then ask the user whether they want to print out the list in the original order, or in reverse. **Extensions:** 1. Enable the user to choose what number item in the list they want to print out 2. Enable the user to only print out a 'slice' of the list (eg item three to item nine only) 3. Enable the user to remove any items of the list that they want to 4. Enable the user to save their list to a file for later, and also enable them to load it back up again too 5. 'Clean' the list by making all the items lowercase.

6	**Max and min list** Write a program that lets the user input a list of numbers. Every time they input a new number, the program should give a message about what the maximum and minimum numbers in the list are. **Extensions** 1. The program should let the user choose the allowable minimum and maximum values, and should not let the user input something to the list that is outside of these bounds 2. The user should be able to write these values to a file and then also read them back out again. 3. If a file has any numbers outside of the boundaries, it should strip them out of the list once it has read them in.
7	**Letter list** Write a program that lets a user choose a letter. The program will then find all the words beginning with that letter in a list and print them out. It should also say how many words it found. **Extensions:** 1. Let the user load up a list of words from a file and have the program process them all 2. Change the program so that the user can choose whether they want all words with only the start of the letter, or ANY place in the word.
8	**RPG character/Pokemon stat creator** Make a program which will randomly create a character's stats based on several rules set by the user. Have it generate a class, gender, strength/magic/dexterity points, and extra abilities or trades. Have the program save it to a file which can then be printed out so that it can be used in a game. **Extension:** 1. Make a mystical name generator. Perhaps randomise different name parts such as sha-ra-lam or big-lim-con to create names for each of your randomly generated characters.
9	**Quiz Maker** Make an application which takes various questions from a file, picked randomly, and puts together a quiz for students, and then reads a key to grade the quizzes. Each quiz can be different.
10	**Check if Palindrome** Checks if the string entered by the user is a palindrome. A palindrome is a word that reads the same forwards as it does backwards like "racecar".

11	**Count Words in a String** Counts the number of individual words in a string. For added complexity, the program could read these strings in from a text file and generate a summary.
12	**Pig Latin** Pig Latin is a game of alterations played on the English language game. To create the Pig Latin form of an English word the initial consonant sound is transposed to the end of the word and an ay is affixed (Ex.: "banana" would yield anana-bay). Read Wikipedia for more information on rules (http://en.wikipedia.org/wiki/Pig_Latin).
13	**Count Vowels** Enter a string and the program counts the number of vowels in the text. For added complexity have it report a sum of each vowel found.
14	**Unit Converter (temperature, currency, volume, mass and more)** 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.
15	**Change Return Program** The user enters a cost and then the amount of money given. You should write a program that works out what denominations of change should be given in pounds, 50p, 20p, 10p etc. **Extensions:** 1. The program will figure out the change for the American currency and the number of quarters, dimes, nickels, pennies needed for the change 2. Make an automatic testing part of your program where it automatically generates a random price and an amount that you give the cashier. It then works out what change to give, and then tests that your program works by adding the change back onto the price of the item to prove your program works. It should flag an error if there are problems.

16	**Shopping list** Create a program that will keep track of items for a shopping list. The program should allow you to keep adding new items. You should also be able to record which shop you are going to visit for the item. **Extensions:** 1. Extend the program to record what priority the item is 2. Extend the program to record whether or not you have bought the item or not 3. Extend the program to say how much you are willing to pay for each item 4. Extend the program to say the quantity needed for each item 5. Make a part of your program that can give you the approximate total of what you intend to spend on your shopping trip. Make sure that you take into account whether you have already bought the item or not (ie don't add this up!).
17	**Hangman** Create a version of the Hangman game using Lists. The program should ask for the word to guess and the number of chances to be given. For every guess you should update the user with which letters they have guessed incorrectly, as well as replacing the letters in the guess word with the ones they have guessed correctly. You should also show the user how many chances they have left. **Extension:** 1. Make sure that you do not let the user lose a life if they guess a letter that they have already guessed.
18	**Squares** Create a program that will ask the user for a number and then print out a list of numbers from 1 to the number entered and the square of the number. For example, if the user entered '3' then the program would output: 1 squared is 1 2 squared is 4 3 squared is 9

19	**Times tables** Create a program which will produce the times table for a number entered by the user eg if the user enters '2' it should produce: 1 x 2 = 2 2 x 2 = 4 3 x 2 = 6
20	**Binary/Hexidecimal/Decimal** Create a program which will convert a given decimal up to 255 into its 8-bit binary equivalent. **Extensions:** 1. Extend program 7 to convert the binary number to hexadecimal 2. Create a program to convert from hexadecimal to decimal.


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.org.uk/gcsereform

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.

A DIVISION OF CAMBRIDGE ASSESSMENT

LLOYD'S REGISTER QUALITY ASSURANCE · ISO9001 · UKAS MANAGEMENT SYSTEMS 001