Pearson Edexcel Level 1/Level 2 GCSE (9–1)

Computer Science

Programming Project

For use from September 2018
Time: 20 hours

You must have: TestPlan_Template.rtf, Test1_Votes.txt and Test2_Votes.txt

Instructions to teachers

- Students should use **one** of the following programming languages:
  - Python
  - Java
  - Pascal/Object Pascal
  - Visual Basic.NET
  - C-derived languages.
- You must adhere to the instructions as specified in the specification.
- Internet access is allowed.
- The materials submitted for assessment must include:
  - evidence of the development of the solution
  - the program code including any necessary solution files
  - a completed programming project authentification form (PPA) – available in the specification – see Appendix 3
  - a completed Head of Centre declaration form which should be signed by the head – available in the specification – see Appendix 4.

Information to students

- The work you submit must be your own.

Turn over ▶

Election

A national sports association is to hold an election for two vacant seats on one of its regional committees. The rules of the association state that the election is to be conducted using a quota-based Single Transferable Vote (STV) system.

The region has 70 members who are eligible to vote. Five candidates are standing for the election, which is to be by postal ballot. An example ballot paper is shown here.

Candidate	Preference
A	2
B	1
C	
D	
E	3

Members complete the ballot paper by indicating their candidate preferences, 1 for their first preference, 2 for their second preference and 3 for their third preference.

The votes from the ballot papers are stored in a text file. The blanks on the ballot paper are recorded as zero.

The association has received 64 valid ballot papers. Four members decided not to vote and two ballot papers had been completed incorrectly.

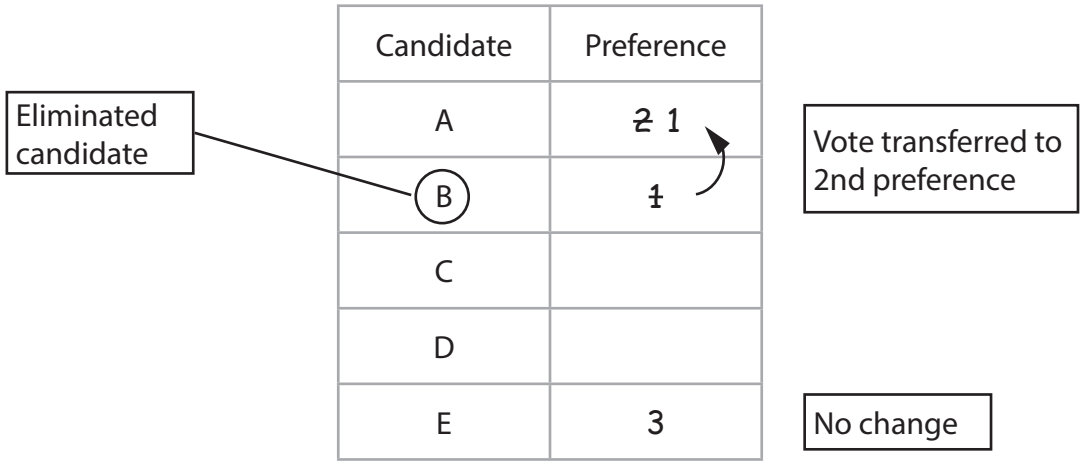
The rules of the association specify the minimum number of first preference votes that a candidate must achieve to be elected. The minimum number is called a quota and is calculated using the formula:

Quota = INTEGER ((valid votes cast / (number of seats +1)) +1)

The following STV system is used when the quota is not achieved by at least two candidates.

Step 1 – The candidate with the fewest first preference votes is eliminated.

Step 2 – Each first preference vote of the eliminated candidate is transferred to the second preference candidate on each of the eliminated candidate's ballot papers. This is illustrated below.



Step 3 – The total of the first preference votes is compared with the quota.

Step 4 – The process is repeated if the quota is not achieved by at least two of the remaining candidates.

If no result is achieved after two candidates have been eliminated and their votes transferred, a re-election is called. The third preference votes are only to be used in the event of a tie.

The association needs a computer program to run the election accurately. The program will read votes from a file, calculate the required quota, apply the STV system and output the results of the election.

Your task is to analyse this problem and to design, implement, test and evaluate a programmed solution. You are provided with two data files. This table describes their purpose and expected outcome.

File Name	Purpose	Expected Outcome
Test1_Votes.txt	No transfer of votes required	Two candidates elected
Test2_Votes.txt	Transfer of votes required	Two candidates eliminated Two candidates elected

The data in Test1_Votes.txt and Test2_Votes.txt should be analysed and used to test your program.

The two files do not contain any invalid ballots. However, if invalid ballots were put into the files, the program should not include them in the STV process.

The Report

Create a folder called Report. Save all your evidence for assessment in this folder. Save your evidence as instructed at each stage.

Stage 1 Analysis (6 marks)

You should include an introduction summarising the overall problem.

The problem should be broken down into sub-problems. You should write a description of each of the sub-problems you identify and explain your selection of sub-problems. State any assumptions you have made.

Save your work in the Report folder as a file called Analysis.

Stage 2 Design (18 marks)

Algorithms (12 marks)

Design algorithms, using pseudo-code or flowcharts, which show a logical solution to each sub-problem. You should include inputs, processes, outputs, validation checks and the programming constructs that you will use when you produce your program.

You should show how the algorithms will link together and lead to an overall solution.

Save your algorithms in the Report folder as a file called Design.

Initial test plan (6 marks)

You should complete the relevant sections of the test plan template provided to produce an initial test plan that will demonstrate your strategy for testing your solution.

Save your initial test plan in the Report folder as a file called TestPlan.

Save a copy of TestPlan in the Report folder as a file called TestTable, to be used in stages 3 and 4.

Stage 3 Implementation (24 marks)

You should translate your design into a program. Ensure that your program is clear and easy to understand.

Add the results of any tests carried out during the implementation stage to the TestTable file.

Save the updated TestTable file.

You should provide evidence showing how you debugged your program. Save your evidence in the Report folder as a file called Debugging.

Create a subfolder called Implementation in the Report folder. Save your source code and all the files required to execute the program in the subfolder.

Stage 4 Testing, refining and evaluation (12 marks)

You should complete the TestTable file by adding any further tests carried out at this stage, including the results of retesting following the correction of any errors.

Save the completed TestTable file.

Evaluate your solution by explaining how well your program meets each of the requirements that you identified in your analysis and describing any refinements that you made to your program during design and implementation.

Save your evaluation in the Report folder as a file called Evaluation.