

**Solution design (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<ul style="list-style-type: none"> <li>• There has been little attempt to decompose the problem.</li> <li>• Some of the component parts of the problem can be seen in the solution, although this will not be complete.</li> <li>• Some parts of the logic are clear and appropriate to the problem.</li> <li>• The use of variables and data structures, appropriate to the problem, is limited.</li> <li>• The choice of programming constructs, appropriate to the problem, is limited.</li> </ul>	<ul style="list-style-type: none"> <li>• There has been some attempt to decompose the problem.</li> <li>• Most of the component parts of the problem can be seen in the solution.</li> <li>• Most parts of the logic are clear and appropriate to the problem.</li> <li>• The use of variables and data structures is mostly appropriate.</li> <li>• The choice of programming constructs is mostly appropriate to the problem.</li> </ul>	<ul style="list-style-type: none"> <li>• The problem has been decomposed clearly into component parts.</li> <li>• The component parts of the problem can be seen clearly in the solution.</li> <li>• The logic is clear and appropriate to the problem.</li> <li>• The choice of variables and data structures is appropriate to the problem.</li> <li>• The choice of programming constructs is accurate and appropriate to the problem.</li> </ul>	<b>3</b>

**Functionality (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.</li> <li>• Program outputs are of limited accuracy and/or provide limited information.</li> <li>• Program responds predictably to some of the anticipated input.</li> <li>• Solution is not robust and may crash on anticipated or provided input.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.</li> <li>• Program outputs are mostly accurate and informative.</li> <li>• Program responds predictably to most of the anticipated input.</li> <li>• Solution may not be robust within the constraints of the problem.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.</li> <li>• Program outputs are accurate, informative, and suitable for the user.</li> <li>• Program responds predictably to anticipated input.</li> <li>• Solution is robust within the constraints of the problem.</li> </ul>	<b>3</b>

**Functionality (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.</li> <li>• Program outputs are of limited accuracy and/or provide limited information.</li> <li>• Program responds predictably to some of the anticipated input.</li> <li>• Solution is not robust and may crash on anticipated or provided input.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.</li> <li>• Program outputs are mostly accurate and informative.</li> <li>• Program responds predictably to most of the anticipated input.</li> <li>• Solution may not be robust within the constraints of the problem.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.</li> <li>• Program outputs are accurate, informative, and suitable for the user.</li> <li>• Program responds predictably to anticipated input.</li> <li>• Solution is robust within the constraints of the problem.</li> </ul>	<b>3</b>

**Solution design (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<ul style="list-style-type: none"> <li>• There has been little attempt to decompose the problem.</li> <li>• Some of the component parts of the problem can be seen in the solution, although this will not be complete.</li> <li>• Some parts of the logic are clear and appropriate to the problem.</li> <li>• The use of variables and data structures, appropriate to the problem, is limited.</li> <li>• The choice of programming constructs, appropriate to the problem, is limited.</li> </ul>	<ul style="list-style-type: none"> <li>• There has been some attempt to decompose the problem.</li> <li>• Most of the component parts of the problem can be seen in the solution.</li> <li>• Most parts of the logic are clear and appropriate to the problem.</li> <li>• The use of variables and data structures is mostly appropriate.</li> <li>• The choice of programming constructs is mostly appropriate to the problem.</li> </ul>	<ul style="list-style-type: none"> <li>• The problem has been decomposed clearly into component parts.</li> <li>• The component parts of the problem can be seen clearly in the solution.</li> <li>• The logic is clear and appropriate to the problem.</li> <li>• The choice of variables and data structures is appropriate to the problem.</li> <li>• The choice of programming constructs is accurate and appropriate to the problem.</li> </ul>	<b>3</b>

**Good programming practices (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<ul style="list-style-type: none"> <li>• There has been little attempt to lay out the code into identifiable sections to aid readability.</li> <li>• Some use of meaningful variable names.</li> <li>• Limited or excessive commenting.</li> <li>• Parts of the code are clear, with limited use of appropriate spacing and indentation.</li> </ul>	<ul style="list-style-type: none"> <li>• There has been some attempt to lay out the code to aid readability, although sections may still be mixed.</li> <li>• Uses mostly meaningful variable names.</li> <li>• Some use of appropriate commenting, although may be excessive.</li> <li>• Code is mostly clear, with some use of appropriate white space to aid readability.</li> </ul>	<ul style="list-style-type: none"> <li>• Layout of code is effective in separating sections, e.g. putting all variables together, putting all subprograms together as appropriate.</li> <li>• Meaningful variable names and subprogram interfaces are used where appropriate.</li> <li>• Effective commenting is used to explain logic of code blocks.</li> <li>• Code is clear, with good use of white space to aid readability.</li> </ul>	<b>3</b>

**Functionality (levels-based mark scheme)**

0	1	2	3	Max.
<i>No rewardable material</i>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.</li> <li>• Program outputs are of limited accuracy and/or provide limited information.</li> <li>• Program responds predictably to some of the anticipated input.</li> <li>• Solution is not robust and may crash on anticipated or provided input.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.</li> <li>• Program outputs are mostly accurate and informative.</li> <li>• Program responds predictably to most of the anticipated input.</li> <li>• Solution may not be robust within the constraints of the problem.</li> </ul>	<p><b>Functionality (when the code is run)</b></p> <ul style="list-style-type: none"> <li>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.</li> <li>• Program outputs are accurate, informative, and suitable for the user.</li> <li>• Program responds predictably to anticipated input.</li> <li>• Solution is robust within the constraints of the problem.</li> </ul>	<b>3</b>