Consider the following algorithm in Fig.2, expressed in pseudocode, as a function $S$:

```pseudocode
function S(A[0..N-1], value, low, high)
    if (high < low)
        return error_message
    mid = (low + high) / 2
    if (A[mid] > value)
        return S(A, value, low, mid-1)
    elseif (A[mid] < value)
        return S(A, value, mid+1, high)
    else
        return mid
endfunction
```

Fig.2

(a) State the name of the algorithm implemented in Fig.2.

(b) Describe the purpose of this algorithm.
Parameters are passed to this function. Complete the following table to identify these parameters and the purpose of each.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Describe what is meant by recursion.

Identify one example of where recursion occurs in this algorithm.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(e) Rewrite the algorithm in Fig.2 without using recursion.

(f) Explain how the algorithm in Fig.2 is an example of a divide and conquer approach.