**Number Representation Practice Task 2**

Binary numbers are expressed as strings comprised entirely of '0's and '1's. Such strings are called binary strings. The flowchart below shows how two binary numbers can be added iteratively.



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions |  |  |  |  |  |  |  |  |
| s1[i] = '0' | Y | Y | Y | Y | N | N | N | N |
| s2[i] = '0' | Y | Y | N | N | Y | Y | N | N |
| carry = '0' | Y | N | Y | N | Y | N | Y | N |
| Action 1 |  |  |  |  |  |  |  |  |
| result = '0'+result | X |  |  | X |  | X | X |  |
| result = '1'+result |  | X | X |  | X |  |  | X |
| Action 2 |  |  |  |  |  |  |  |  |
| carry = '0' | X | X | X |  | X |  |  |  |
| carry = '1' |  |  |  | X |  | X | X | X |

**NOTE:** Here, padding means to add '0's at the beginning of the shorter string so that both strings become the same length.

For example, if s1 = '10' and s2 = '1001'

then padding changes s1 to '0010'.

**Task 2.1**

Write Python code to implement the algorithm in the flowchart as a function add\_bin\_iter(s1,s2).

[7]

The flowchart below is for a function, add\_bin\_recur(s1,s2) that adds two binary numbers recursively.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Conditions |  |  |  |  |  |  |
| s1[0] = '0' | Y | N | Y | Y | N | N |
| s2[0] = '0' | Y | N | N | N | Y | Y |
| s[0] = '0' | – | – | Y | N | Y | N |
| Action |  |  |  |  |  |  |
| OUTPUT '0'+s | X |  |  |  |  |  |
| OUTPUT '1'+s |  | X |  |  |  |  |
| OUTPUT '01'+s[1:] |  |  | X |  | X |  |
| OUTPUT '10'+s[1:] |  |  |  | X |  | X |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Conditions |  |  |  |  |
| s1 = '0' | Y | Y | N | N |
| s2 = '0' | Y | N | Y | N |
| Action |  |  |  |  |
| OUTPUT '00' | X |  |  |  |
| OUTPUT '01' |  | X | X |  |
| OUTPUT '10' |  |  |  | X |

**Task 2.2**

Write Python code to implement the function add\_bin\_recur(s1,s2). [6]

**Task 2.3**

Write Python code that does the following:

* Prompt the user to input a binary number.
* Validate the input.
* Prompt the user to input a second binary number.
* Validate the second input.
* Use either of the algorithms in the earlier tasks to add the two binary numbers that were given by the user. Note that both algorithms may output a string that begins with '0'. Your code should remove any initial '0's before displaying the output to the user.

[7]

**Task 2.4**

Write Python code to convert a binary number, given as a binary string, into an integer displayed in denary. You should not use the int command in Python.

[4]