**AutoBench™ Version 1.1**

**Benchmark Name: Pointer Chasing**

**Benchmark Description**
This EEMBC benchmark simulates an embedded automotive/industrial application which performs a lot of pointer manipulation.

The kernel employs a doubly linked list then searches the list for entries which match an input token. A large set of input tokens is used to exercise the entire list. The number of steps taken to find each input token is recorded.

**Optimization Rules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI C</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Intrinsics/Language Extensions</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Custom Libraries</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Assembly Language</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HW Accelerators</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Algorithm Flowchart**

[Flowchart showing the algorithm steps]

Start

- Initialize and Get Test Data

- Get Input Token

- Search Doubly Linked List

- Save Steps Token

- Check: done?
  - No: Loop back
  - Yes: Closing, Report Results

Stop