**说明文档**

|  |  |  |
| --- | --- | --- |
| 马九思 | 12335158 | 12统计 |
| 张超逸 | 12335269 | 12统计 |
| 王政纲 | 12335202 | 12统计 |
| 刘盈 | 12335146 | 12统计 |

我们根据师兄的五阶段程序，设计了本次试验。

首先，我们定义了一个“two”的primitive\_function，在栈的末尾添加2。

*define\_primitive\_function "two", two*

*;; << -- 2 >>*

*xor rax, rax*

*inc rax*

*inc rax*

*push\_argument\_stack rax*

*next*

然后，定义一个构造连乘的数列的函数Mul\_series。

*define\_function "Mul\_series",Mul\_series*

*xx one,two*

*xx xswapxx*

*xx help\_Mul\_series*

*xx drop*

*xx end*

我们将1，2，…入栈，使用one，two函数构成数列前两项，再用x|swap|xx函数把输入的最后一个数调整到栈顶，以便进行乘法运算。

为了避免函数过于复杂，我们还要定义一个使用递归算法的辅助函数help\_Mul\_series帮助实现运算。

*define\_function "help\_Mul\_series", help\_Mul\_series*

*xx dup, one?, false?branch,2*

*xx end*

*xx sub1*

*xx xxoverx*

*xx multiple*

*xx swap*

*xx taca, help\_Mul\_series*

函数使用dup函数复制栈顶数值one?，false?branch,2进行判断 ，为了不改变原来的序列，使用xx|over|x函数调整数列，在让末尾两数相乘合并，再swap末尾二数，为递归做准备。直到one?, false?branch,2进行判断结束，end。返回到Mul\_series函数中，此时末尾的数值是判断成功后的1，所以最后用drop把栈顶1去掉，以结束该函数运行 。

结果截图：



