**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÁO CÁO BÀI TẬP LỚN MÔN KIẾN TRÚC MÁY TÍNH**

**GIÁO VIÊN: TRẦN THANH BÌNH LỚP L05**

**Problem 2: Nhân 2 số thực**

THÀNH VIÊN:

TRẦN QUANG KHẢ - 1913764

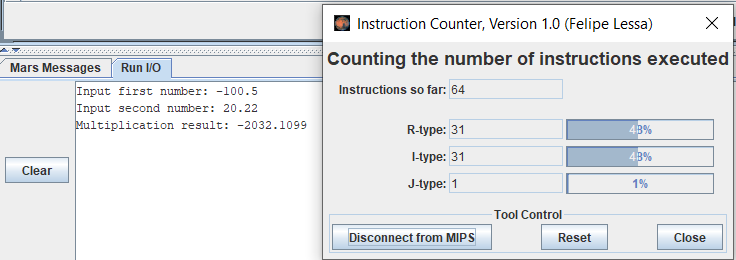
LÊ HỒNG NHẬT - 1914474

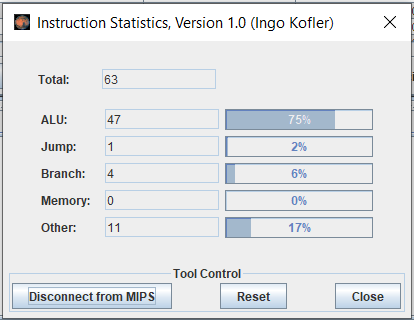
* Thống kê số lệnh/loại lệnh sử dụng trong chương trình.

Khi chạy chương trình với tescase 5:

**Phép tính: -100.5 \* 20.22 = -2032.11**

Ta thu được bảng thống kê như sau:





* Tính và trình bày các tính thời gian chạy của chương trình trên máy tính MIPS có tần số 2GHz

Ta có CPI mỗi lệnh = 1.

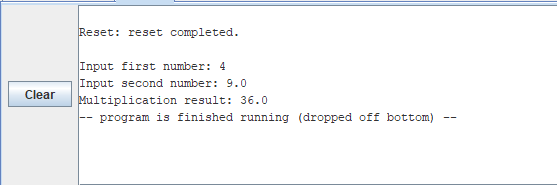
Time execution = (CPI\*IC)/clockrate

Time execution = (1\*64)/(2\*10^9) = 3.2\*10^-8 (s)

Testcase 1:

Phép tính trên thực tế: 4 \* 9.0 = 36.0.

Kết quả phép tính khi sử dụng chương trình:

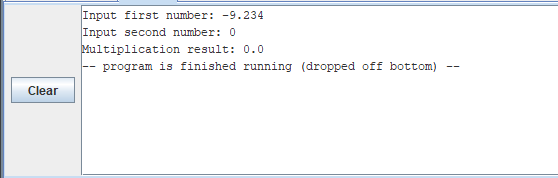


Testcase 2:

Phép tính trên thực tế: -9.234 \* 0 = 0

Kết quả phép tính khi sử dụng chương trình:

Trong chương trình nếu số nhân hoặc số bị nhân bằng 0 thì chương trình sẽ xuất ra kết quả là 0 mà không cần phải tính toán gì cả.

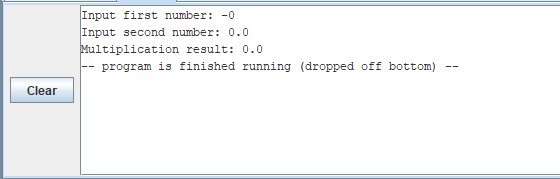


Testcase 3:

Phép tính trên thực tế: -0 \* 0.0 = 0

Kết quả phép tính khi sử dụng chương trình:

Trong chương trình nếu số nhân hoặc số bị nhân bằng 0 thì chương trình sẽ xuất ra kết quả là 0 mà không cần phải tính toán gì cả.

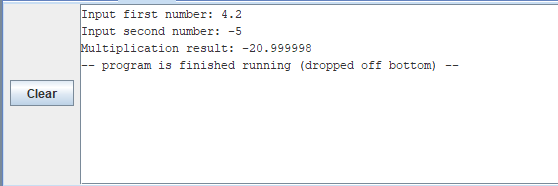


Testcase 4:

Phép tính trên thực tế: 4.2 \* -5 = -21.

Kết quả phép tính khi sử dụng chương trình:

Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.

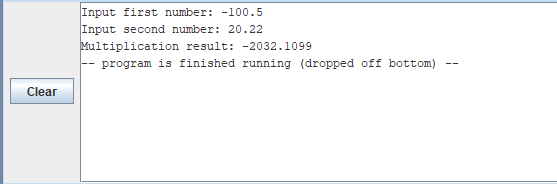


Testcase 5:

Phép tính trên thực tế: -100.5 \* 20.22 = -2032.11

Kết quả phép tính khi sử dụng chương trình:

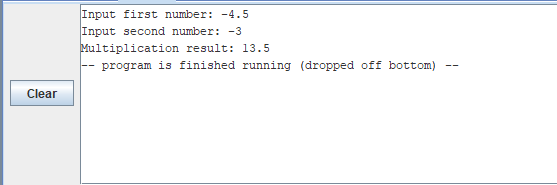
Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.



Testcase 6:

Phép tính trên thực tế: -4.5 \* -3 = 13.5

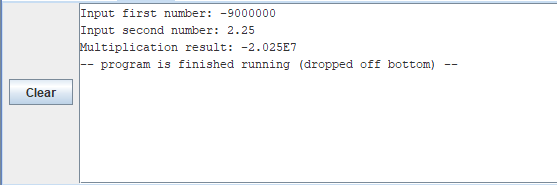
Kết quả phép tính khi sử dụng chương trình:



Testcase 7:

Phép tính trên thực tế: -9000000 \* 2.25 = -20250000

Kết quả phép tính khi sử dụng chương trình:

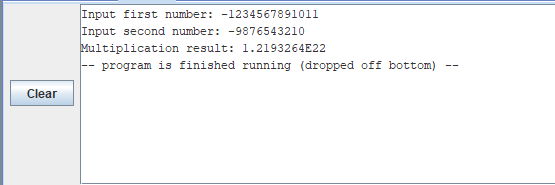


Testcase 8:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): -1234567891011 \* -9876543210 = 1.219326312\*10^22.

Kết quả phép tính khi sử dụng chương trình:

Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.

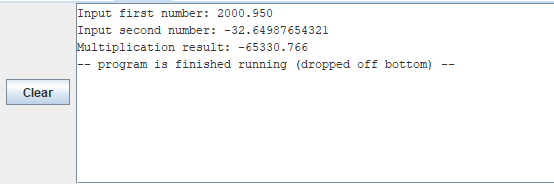


Testcase 9:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): 2000.950 \* -32.64987654321 = -65330.77047

Kết quả phép tính khi sử dụng chương trình:

Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.

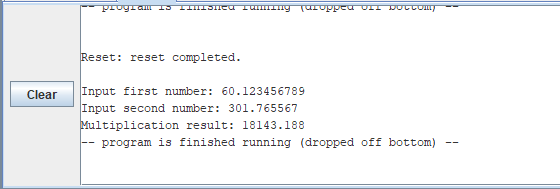


Testcase 10:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): 60.123456789 \* 301.765567 = 18143.18903

Kết quả phép tính khi sử dụng chương trình:

Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.

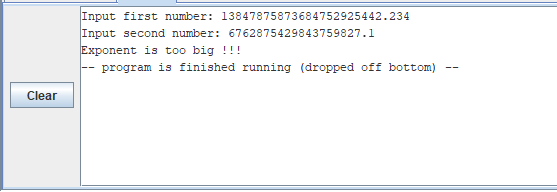


Testcase 11:

Phép tính trên thực : 13847875873684752925442.234 \* 6762875429843759827.1 = (số rất lớn)

Kết quả phép tính khi sử dụng chương trình:

Vượt quá giới hạn của số thực 32 bit. Khi ta phân tích thì phần mũ của các số hạng chiếm hơn 9 bit, nên ở đây chương trình sẽ xuất ra màn hình:



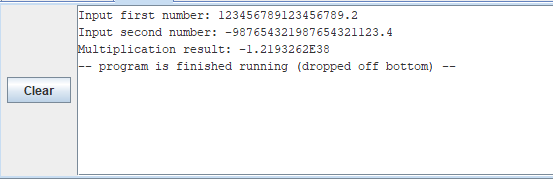
Testcase 12:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX):

123456789123456789.2 \* -987654321987654321123.4 = -1.2193263\*10^38

Kết quả phép tính khi sử dụng chương trình:

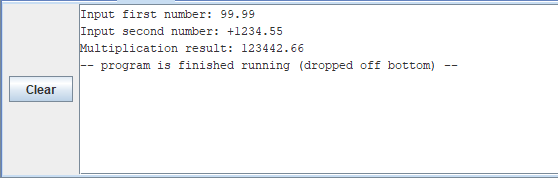
Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế.



Testcase 13:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): 99.99 \* +1234.55 = 123442.6545

Kết quả phép tính khi sử dụng chương trình:

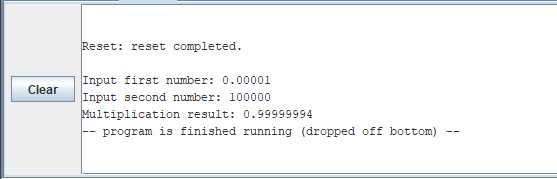


Testcase 14:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): 0.00001 \* 100000 = 1

Kết quả phép tính khi sử dụng chương trình:

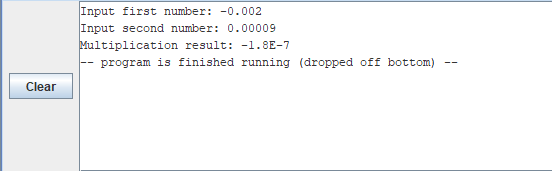
Chương trình sẽ cho ra kết quả gần đúng so với kết quả thực tế



Testcase 15:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX): -0.002 \* 0.00009 = -1.8\*10^-7

Kết quả phép tính khi sử dụng chương trình:



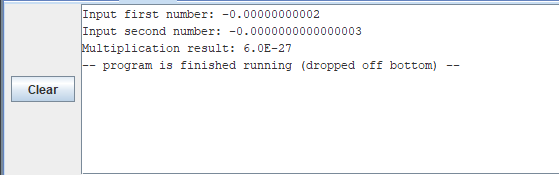
Testcase 16:

Phép tính trên thực tế (sử dụng CASIO fx-580VNX):

-0.00000000002 \* -0.0000000000000003 = 6\*10^-27

Trên thực tế thì với các số quá nhỏ thì máy tính CASIO fx580VNX cũng cho ra kết quả xấp xỉ.

Kết quả phép tính khi sử dụng chương trình:



Testcase 17:

Số bị nhân hoặc số nhân là số quá nhỏ. Khi đó 8 bit exponent sẽ bị tràn.

0.000000000000000000000134 \* 0.00000000000000000000000000034 = (số quá nhỏ)

