ICS13' Homework 2

1. SEQ:
   Now we have following Y86 code segment:

```
1 0x000: irmovl $9, %edx
2 0x006: irmovl $21, %ebx
3 0x00c: subl %edx, %ebx
4 0x00e: irmovl $128,%esp
5 0x014: rmmovl %esp, 100(%ebx)
6 0x01a: pushl %edx
7 0x01c: popl %eax
8 0x01e: je done
9 0x023: call proc
10 0x028: done:
11 0x028: halt
12 0x029: proc:
13 0x029:   ret
```

   Please write down the trace of executing line 9 "**call proc**" (both Generic and Specific).

2. PIPE Basic:
   Ann, Brian, Cathy, Dave each have one load of clothes to **wash**, **dry**, and **fold**. The Machine Washer takes 30 minutes, Dryer takes 40 minutes, and Magic "Folder" takes 20 minutes to deal with one load of clothes. If they worked sequentially, how long would the whole process take? What if they used Pipelining? Please show me the Speedup (old/new).

3. Hazard Basic:
   In PIPE Implementation, How many kinds of Hazards existed? And what are the causes?

4. Data Hazard:
   Now we have following Y86 code segment:

```
addl %ebx, %eax
mrmovl 10(%eax), %ecx
mrmovl 20(%ecx), %ecx
addl %ecx, %ebx
```

   They are executed on PIPE, Mark all of the data hazards in the above code.

5. Forwarding:

```
#prog1
irmovl $10,%edx
irmovl $20,%eax
addl %edx,%eax
halt
```

```
#Prog2
addl %eax,%ebx
addl %edx,%eax
```

1) How to solve above data hazard in prog1 by Forwarding ? (detail required)
2) In Prog2, Is there a Data Hazard? If yes, show how to solve it. If not, why ?

6. Open Question:
   We have learnt about Pipelining, Execute instructions by five stages: Fetch \Decode \Execute\ Memory\ and Write Back. We do that because we want speedup. Someone said that **Number of Pipe Stages** leads to Potential Speedup. Indeed Intel I3-540 Processor has 16-stage pipelines, and Prescott Processor may have more than 30 stages. What's your opinion? More Stages mean More Speedup? (This is Open Question. You can search for some reference to write down your own answer. And Enjoy your holiday, See you~)