Exercise 4:

Write a Pintool in JIT mode and in Probe mode.

The pintool receives 2 possible knobs: "**-prof**" and "**-inst**" that are to be applied as follows:

1. **<pindir>/pin –t ex4.so –prof -- ./bzip2 –k –f input.txt**
2. **<pindir>/pin –t ex4.so –inst -- ./bzip2 –k –f input.txt**

When applied with the "**-prof**" knob the pintool should run exercise 2 and print out loop trip count information into the file "**loop-count.csv**" according to the instructions of exercise 2.

When applied with the "**-inst**" knob the pintool should run in probe mode and generate the binary code of the top 10 routines according to the gathered profiling data from previous run as in exercise 3 and apply loop unrolling by 4 for the following loop in routine "**fallbackSort**" starting at address **0x409fde** and ending at address **0x40a076**:

```
409fde:   mov    -0x18(%rbp),%eax
409fe1:   sar    $0x5,%eax
409fe4:   cltq
409fe6:   lea    0x0(,%rax,4),%rdx
409fee:   mov    -0x868(%rbp),%rax
409ff5:   add    %rdx,%rax
409ff8:   mov    (%rax),%edx
409ffa:   mov    -0x18(%rbp),%eax
409ffd:   and    $0x1f,%eax
40a000:   mov    $0x1,%esi
40a005:   mov    %esi,%ebx
40a007:   mov    %eax,%ecx
40a009:   shl    %cl,%ebx
40a00b:   mov    %ebx,%eax
40a00d:   and    %edx,%eax
40a00f:   test   %eax,%eax
40a011:   je     40a019 <fallbackSort+0x3e6>
40a013:   mov    -0x18(%rbp),%eax
40a016:   mov    %eax,-0x1c(%rbp)
40a019:   mov    -0x18(%rbp),%eax
40a01c:   cltq
40a01e:   lea    0x0(,%rax,4),%rdx
40a026:   mov    -0x858(%rbp),%rax
40a02d:   add    %rdx,%rax
40a030:   mov    (%rax),%edx
40a032:   mov    -0x14(%rbp),%eax
40a035:   mov    %edx,%ecx
40a037:   sub    %eax,%ecx
40a039:   mov    %ecx,%eax
40a03b:   mov    %eax,-0x20(%rbp)
40a03e:   cmpl   $0x0,-0x20(%rbp)
40a042:   jns    40a04d <fallbackSort+0x41a>
40a044:   mov    -0x86c(%rbp),%eax
40a04a:   add    %eax,-0x20(%rbp)
40a04d:   mov    -0x20(%rbp),%eax
40a050:   cltq
40a052:   lea    0x0(,%rax,4),%rdx
40a05a:   mov    -0x860(%rbp),%rax
40a061:   add    %rax,%rdx
40a064:   mov    -0x1c(%rbp),%eax
```

```
40a067:    mov     %eax,(%rdx)
40a069:    addl    $0x1,-0x18(%rbp)
40a06d:    mov     -0x18(%rbp),%eax
40a070:    cmp     -0x86c(%rbp),%eax
40a076:    jl      409fde <fallbackSort+0x3ab>
```

Place the translated routines in an allocated memory area and patch them to the original image code.

For the exercise it is recommended to use the provided pintool source code "**rtn-translation-pin3.cpp**" located at:

https://moodle.technion.ac.il/mod/resource/view.php?id=461221

In both modes **–prof** and **–inst** the pintool should not run more than 30% slower compared to the same run of the **bzip2** using pin but without the ex4.so pintool, as measured by the 'time' command.

Test your pintool:

In the moodle you'll find the input binary file called "**bzip2.gz**" along with an input file to give it called "**input.txt.gz.**
Ftp the files to your Linux account and open them using the **gunzip** command.
To run it simply type: $ **./bzip2 –k –f input.txt**
This will compress the file **input.txt** and generate a new file **input.txt.bz2**

To test your pintool on the above **bzip2** binary file, simply type:
$ time <pindir>/pin –t ex4.so <-prof/-inst> -- ./bzip2 –k –f input.txt

Submission requirements:
The submission of this exercise is **in pairs only**.
Submit 1 compressed file called **"ex4.zip"** into the moodle exercise 4 link containing the following files:

1. The binary of your pintool **ex4.so** (compiled, and tested by you that it runs and gives the result).
2. A directory called: 'src' containing all the sources of your pintool along with the make files and a REDAME.txt file that includes the following:
    a. names + id numbers
    b. compilation command
    c. how to run the tool.

Submission deadline: midnight Sunday July 5, 2018.