Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 5 REPORT

Taylan ÖNDER
151044015

Course Assistant: Özgü GÖKSU

1 INTRODUCTION

1.1 Problem Definition

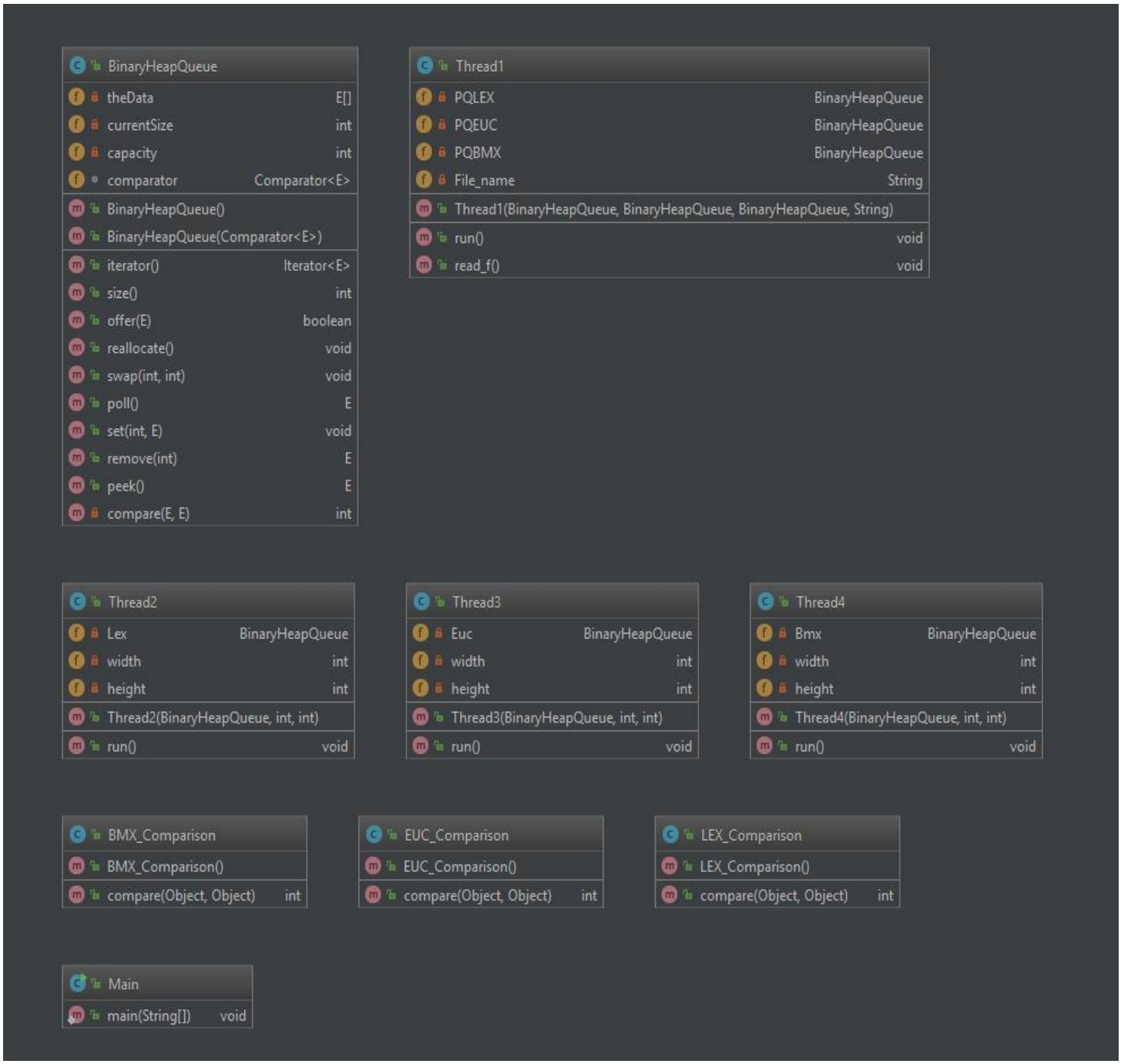
According to the given image using priority queue (own implementation) with various comparison types (Lex,Euc and Bmx) images will load and their pixels color printed according to specific priorities. A mechanism that works at the same time with different threads is required using 3 different queues which sort according to priority com

1.2 System Requirements

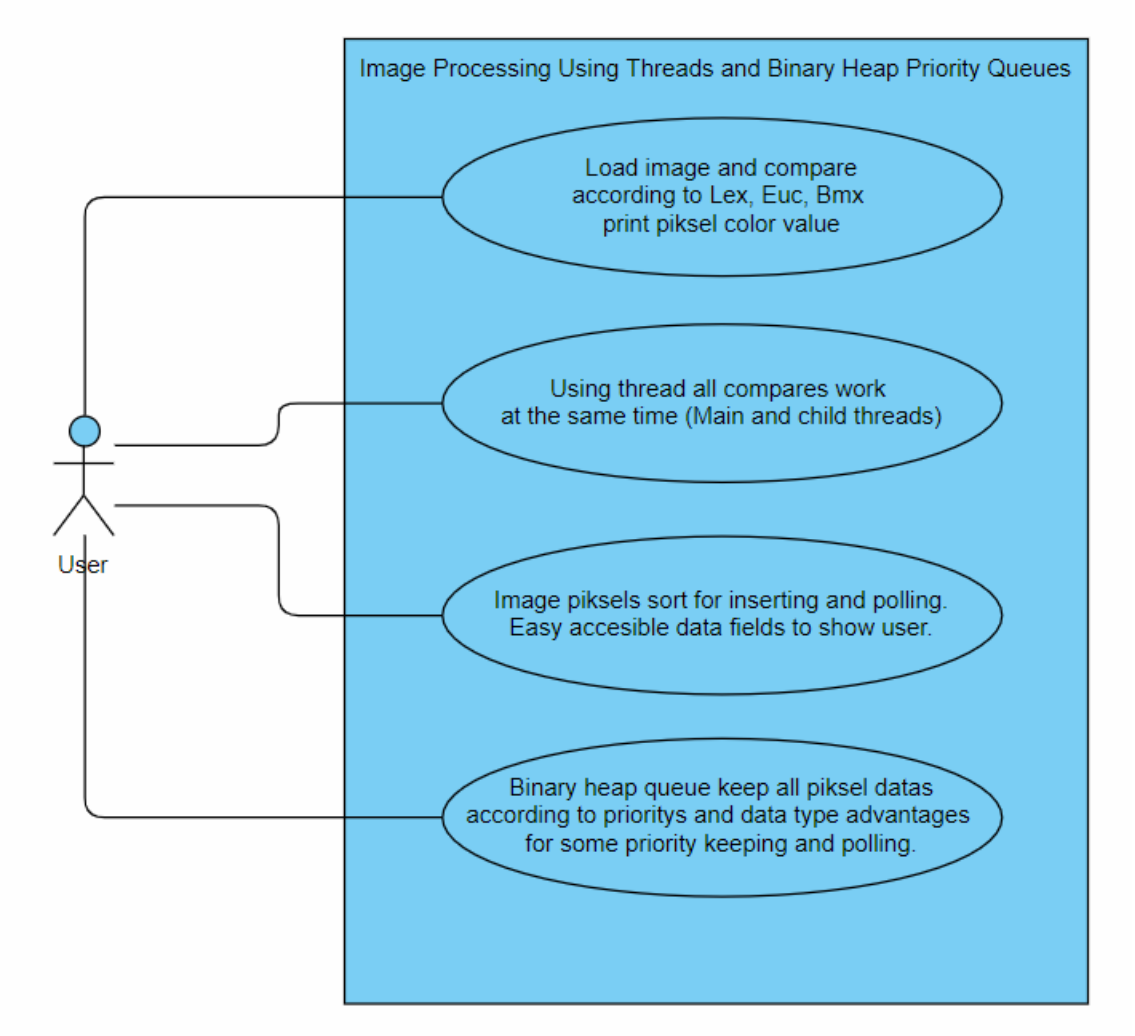
We need 3 priority queues (own implementation binary heap priority queue class) to keep piksel of images and according to compare types (Lex, Euc and Bmx) offer and pool methods. We need three thread class to overload 4 run methods to work at the same time.We need 3 different comparable class to overload compare method according to compare types offer and poll method to use in binary heap queue.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagram



2.2 Problem Solution Approach

Solution of my problem is using threads and binary heap priority queues.First of all I create first thread to read image to convert pixsels and keep count to start other threads.First thread read image convert pixels and insert 3 different priority queues according to compare types.If first thread count equals 100 then start also 3 different thread to poll pixels in queues and print colors of pixels.I have priority queue for insert and poll element in threads according to compare types.I overload compare methods for 3 compare types.When inserting or polling pixels in queue, methods use compare methods different 3 overloading compare methods in 3 different class.I use synchronise method to avoid overlapping of threads.To solve producer consumer problem, I use synchronise method and check according to queue is empty or not, I used wait in thread to other threads running and offering elements empty queues.When for loop turn size of image width*height,thread1 ends and all threads terminate. Thus the program terminates safely.

3 RESULT

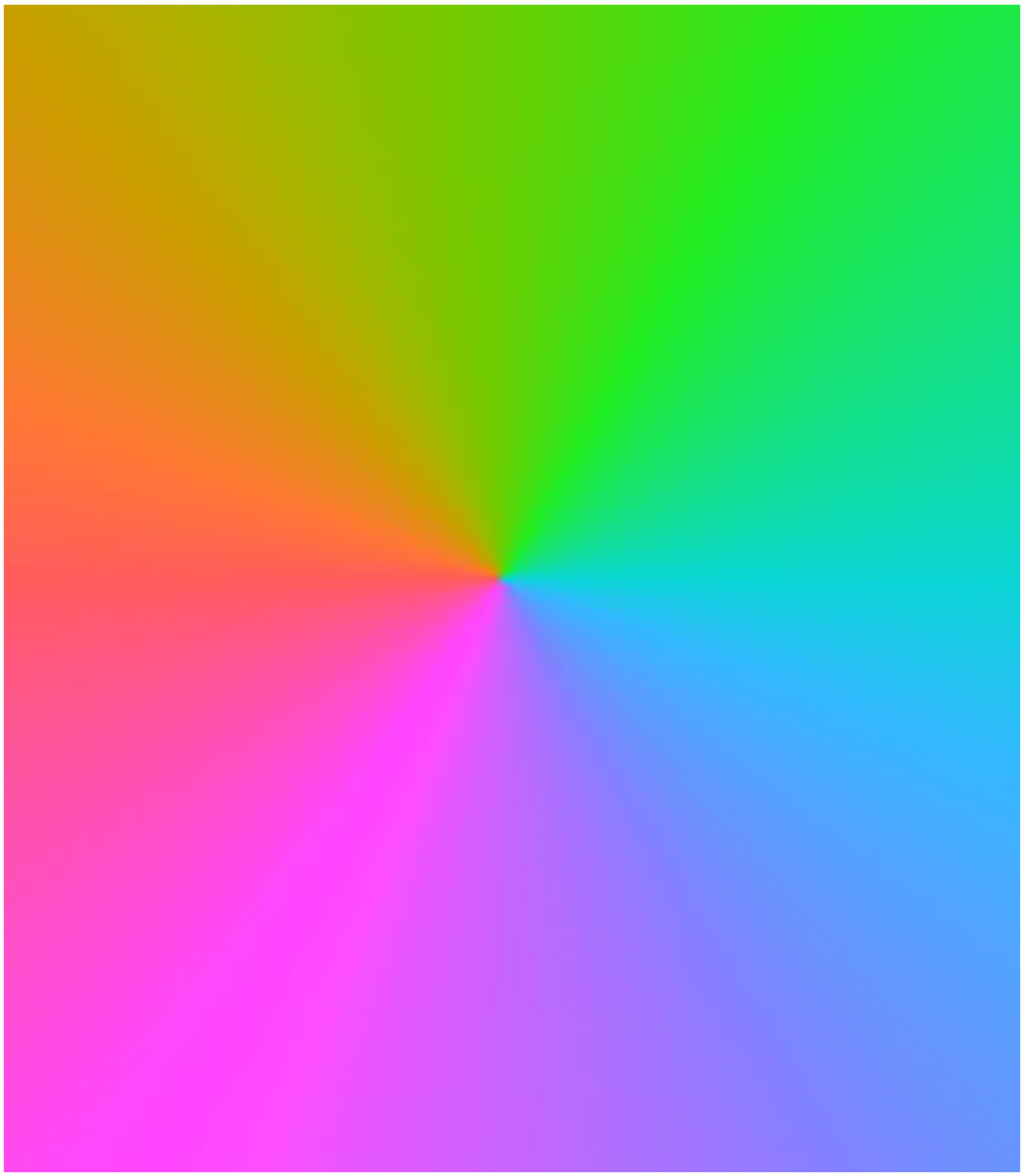
3.1 Test Cases

Part 1

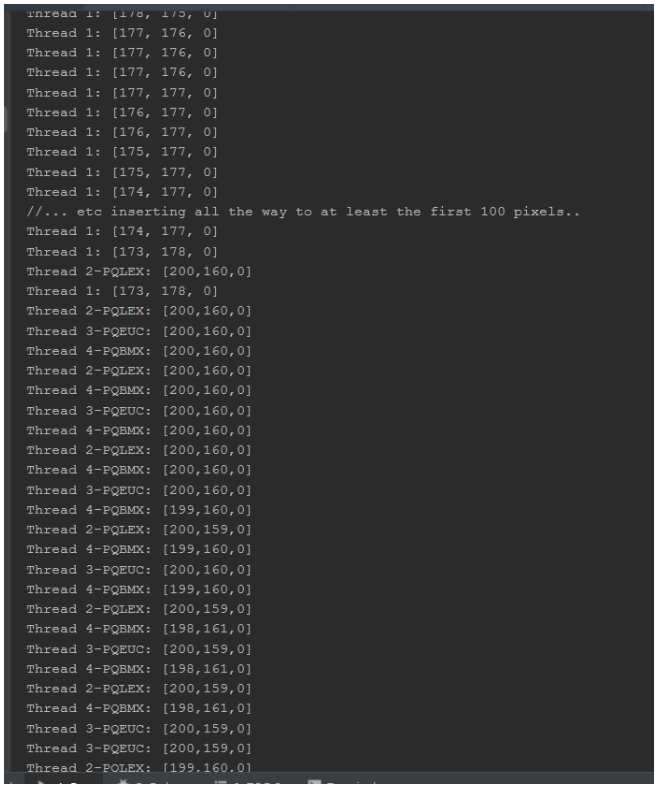
1. Firstly loading image and find width and column value to loop times.
2. Secondly create compare objects.
3. Thirdly in thread1 insert pixels in 3 different queues according to priority (compare types).
4. Fourthly keep count and when count equals 100 start other 3 threads to poll pixels and print screen.
5. Fifthly when turn image width* image height program and all thread terminated.

3.2 Running Results

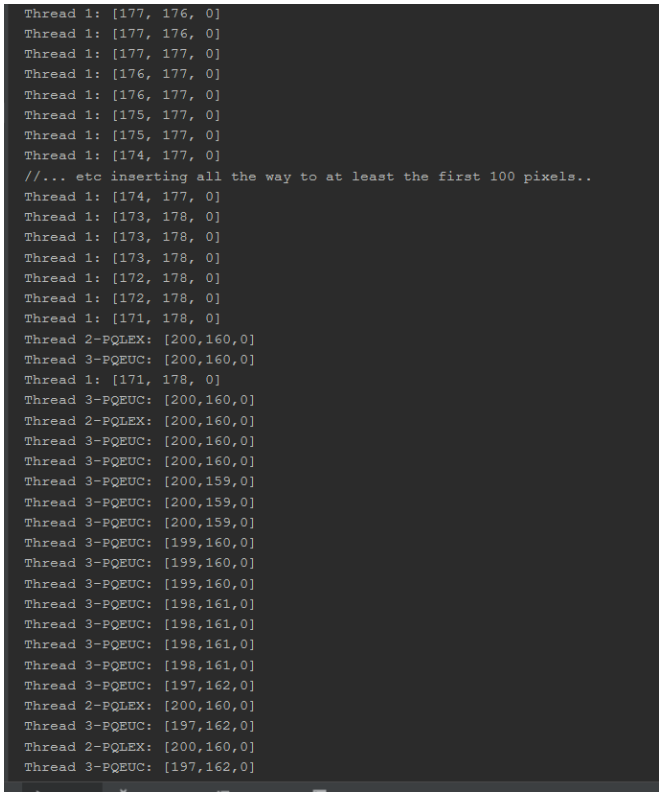
Input -> Image



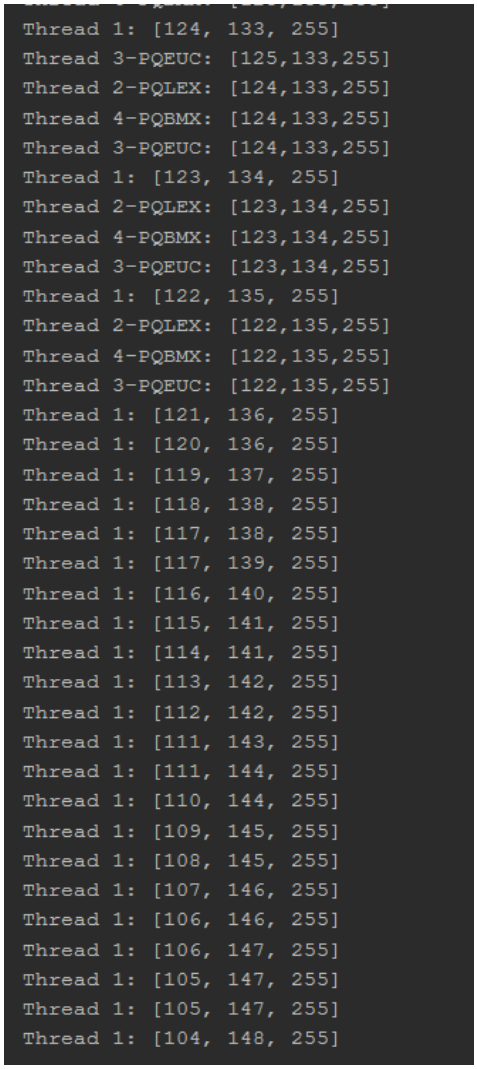
Output

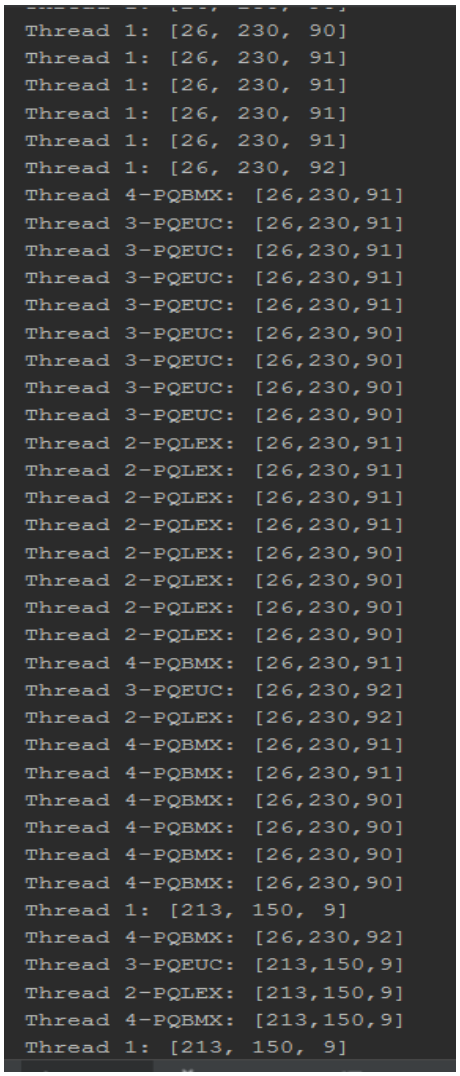


After first 100 pixel offering start thread2,thread3 and thread4.

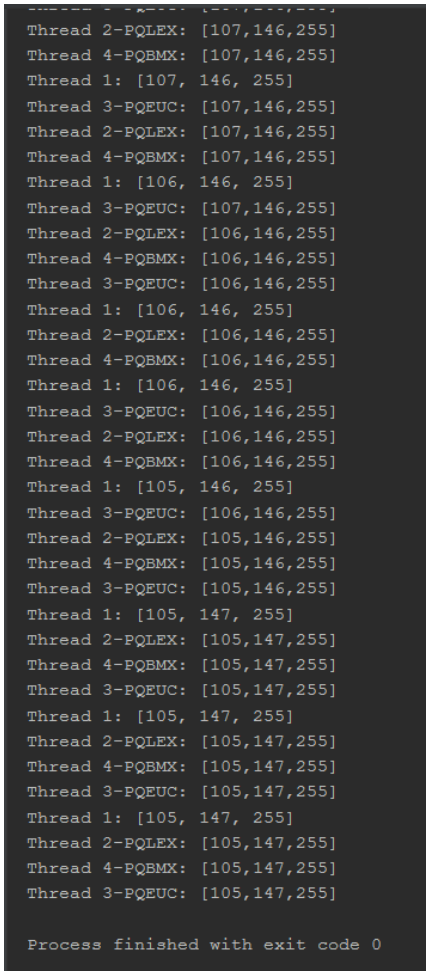


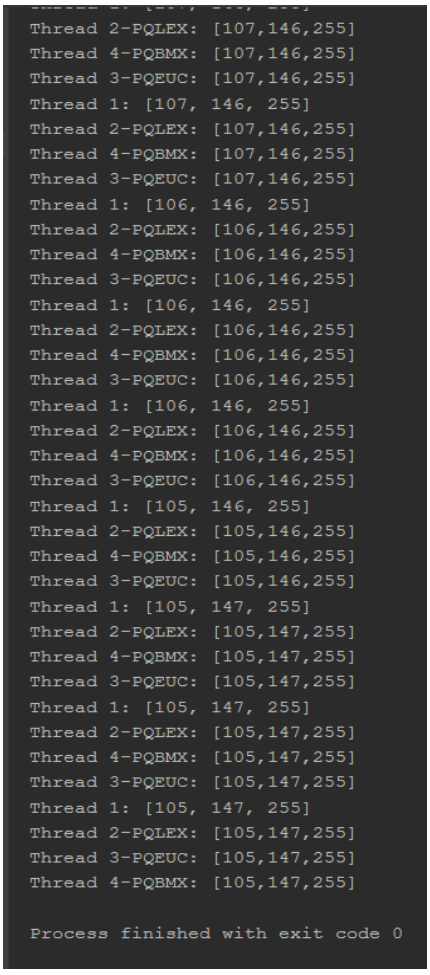
Other after first 100 pixel offering start thread2,thread3 and thread4.





A sections from the middle of the running program.





A sections from the terminated program.