Name: Terje Haugum
Subject: Programmeringsspråk

Due: 22.10.2020 23:59
Assignment: 4

Assignment 4

Task 3(Multi-threading):

The first three digits of running task 3 is: 100

When running with two separate threads has some advantages regarding the CPU. It will be able to compute “parallel” calculations. This improves the performance and prevents race conditions where the threads “compete” about the shared data as the shared data is locked.

Task 4 (Theory Lazy):

Using lazy evaluation, the statement is only executed when the result is needed elsewhere in the code. In our task and in the code delivered you can see the ‘
thread lazyOdds = {GenerateOddsLazy . .} end ‘ will create a “Stopped execution”. The elements in the list generated by the function are only accessed when needed elsewhere in the code (When needed by the Product-function).

Being “lazy” the list generated is restricted by its needs (Product), and it will not generate anything unless it is requested. This implies low use of recourses and an increased throughput. In contrast, using {GenerateOdd} without lazy would be able to use as many recourses as possible to fulfill its task. This could cause high recourse usage.

Task 5c (BoundedBuffer)

Bounded buffer is a combination of lazy and eager execution. In the task we create a buffer that stores N amount of Hammers so the buffer is therefore bounded by N. In the function we need to create a thread that looks at the end of the stream of hammers in order for it to be ready for a new order when needed. Whenever there is need of more hammers, an inner function is called.

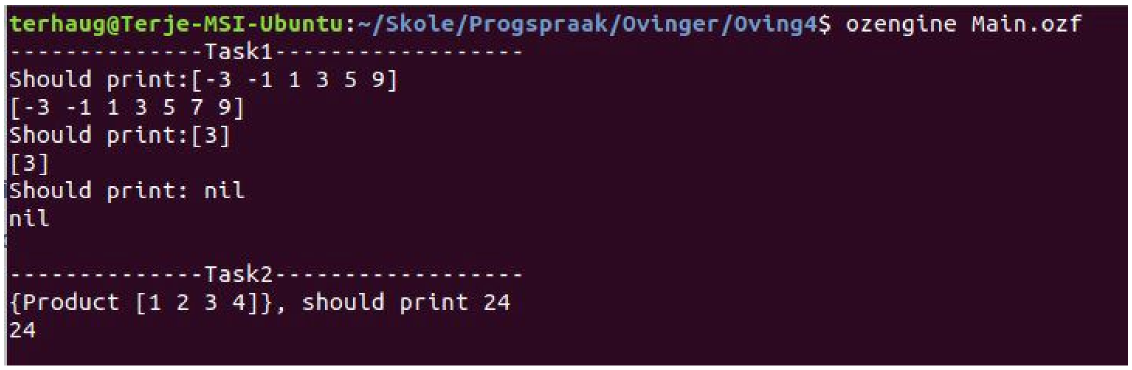
There is an inner part here where you need to return the Head of the stream at the same time produce a new hammer that I could not figure out.

Finally the buffer is returned.

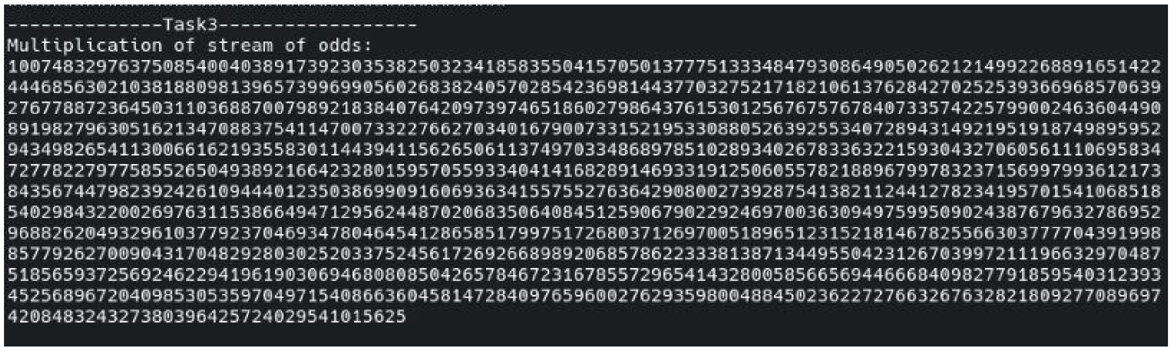
Name: Terje Haugum
Subject: Programmeringsspråk

Due: 22.10.2020 23:59
Assignment: 4

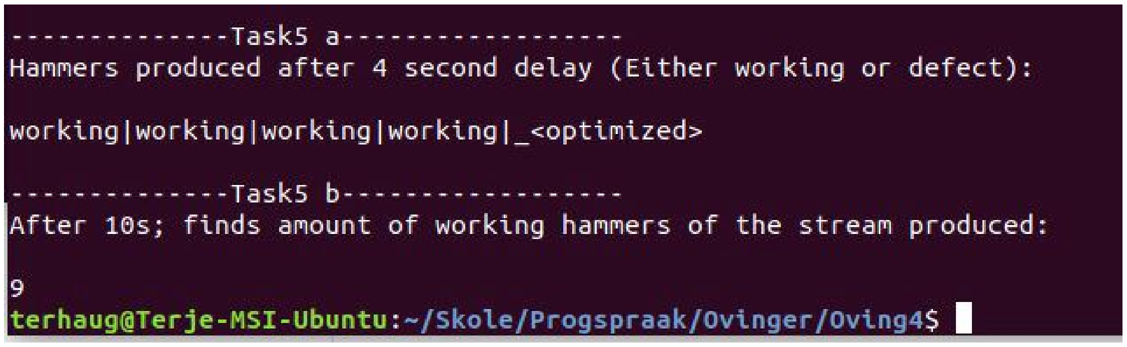
Outputs:



```
terhaug@Terje-MSI-Ubuntu:~/Skole/Progspraak/Ovinger/Oving4$ ozengine Main.ozf
--------------Task1------------------
Should print:[-3 -1 1 3 5 9]
[-3 -1 1 3 5 7 9]
Should print:[3]
[3]
Should print: nil
nil

--------------Task2------------------
{Product [1 2 3 4]}, should print 24
24
```



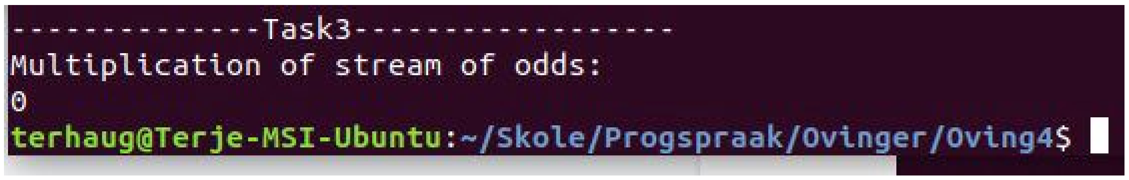
```
--------------Task3------------------
Multiplication of stream of odds:
1007483297637508540040389173923035382503234185835504157050137775133348479308649050262121499226889165142 2
4446856302103818809813965739969905602683824057028542369814437703275217182106137628427025253936696857063 9
2767788723645031103688700798921838407642097397465186027986437615301256767576784073357422579900246360449 0
8919827963051621347088375411470073322766270340167900733152195330880526392553407289431492195191874989595 2
9434982654113006616219355830114439411562650611374970334868978510289340267833632215930432706056111069583 4
7277822797758552650493892166423280159570559334041416828914693319125060557821889679978323715699799361217 3
8435674479823924261094440123503869909160693634155755276364290800273928754138211244127823419570154106851 8
5402984322002697631153866494712956244870206835064084512590679022924697003630949759950902438767963278695 2
9688262049329610377923704693478046454128658517997517268037126970051896512315218146782556630377770439199 8
8577926270090431704829280302520337524561726926689892068578622333813871344955042312670399721119663297048 7
5185659372569246229419619030694680808504265784672316785572965414328005856656944666840982779185954031239 3
4525689672040985305359704971540866360458147284097659600276293598004884502362272766326763282180927708969 7
420848324327380396425724029541015625
```



```
--------------Task5 a------------------
Hammers produced after 4 second delay (Either working or defect):

working|working|working|working|_<optimized>

--------------Task5 b------------------
After 10s; finds amount of working hammers of the stream produced:

9
terhaug@Terje-MSI-Ubuntu:~/Skole/Progspraak/Ovinger/Oving4$
```

FYI:
I had some issues when running task 3 and 4 on my ubuntu 18.04 installed on my laptop. It returned the result 0. Running the same code on a different machine resulted in the output seen above. Do you have any answers to what may cause this?

Here is output from ubuntu:



```
--------------Task3------------------
Multiplication of stream of odds:
0
terhaug@Terje-MSI-Ubuntu:~/Skole/Progspraak/Ovinger/Oving4$
```

Task 4 is not included as it produces same result as task 3.