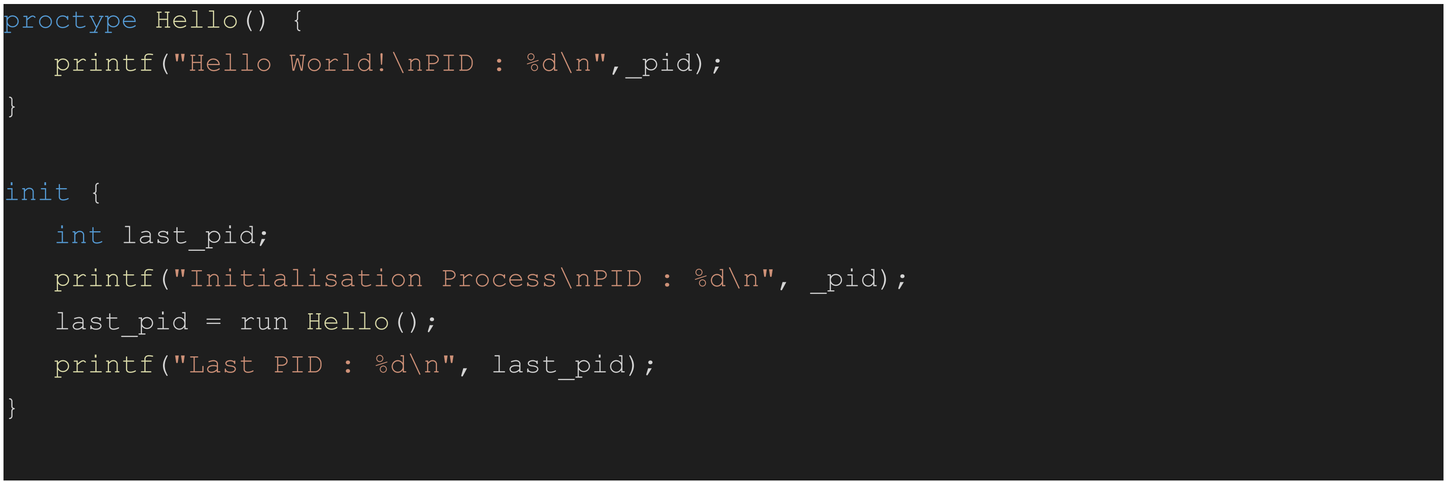
# Software Engineering

## Assignment 4

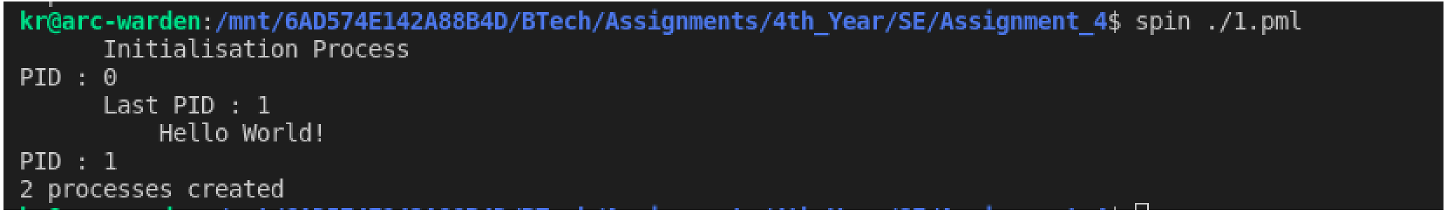
Student Details

Name: Krunal Rank
Adm. No: U18CO081

1. Write a program to create a process that prints “Hello World”. Use run in init process to instantiate it and _pid to print the ids of all created processes.

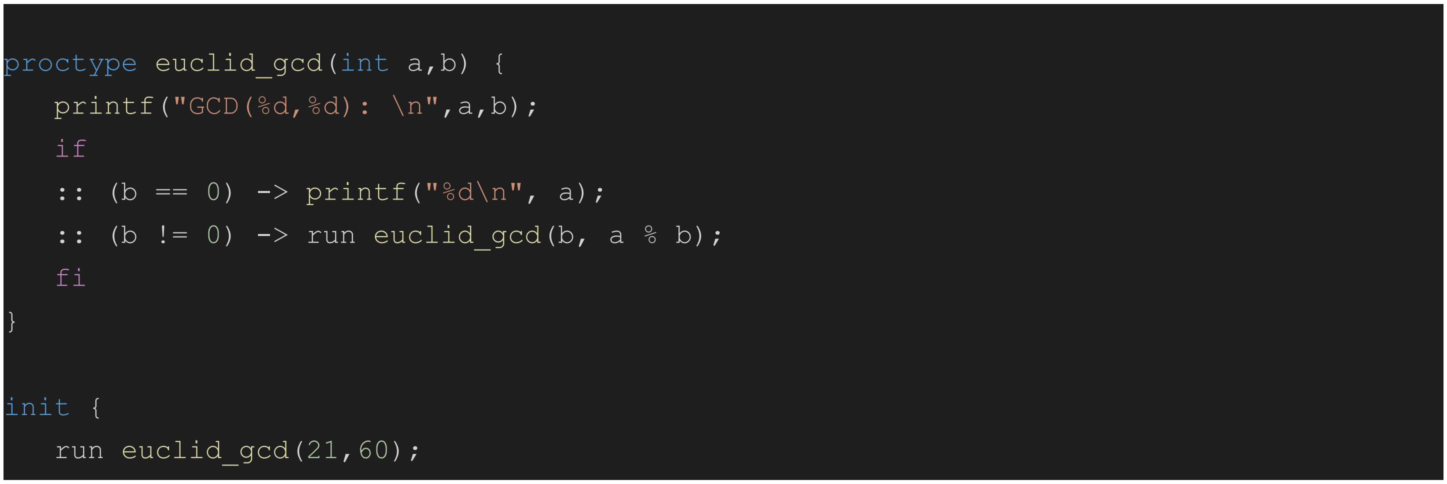


```
proctype Hello() {
   printf("Hello World!\nPID : %d\n",_pid);
}

init {
   int last_pid;
   printf("Initialisation Process\nPID : %d\n", _pid);
   last_pid = run Hello();
   printf("Last PID : %d\n", last_pid);
}
```



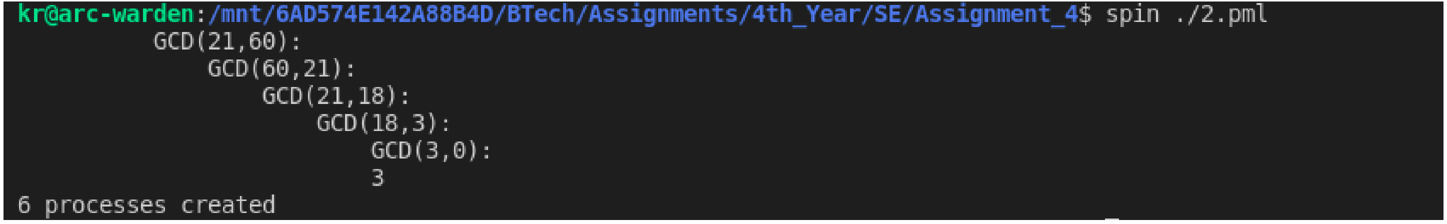
```
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/SE/Assignment_4$ spin ./1.pml
      Initialisation Process
PID : 0
      Last PID : 1
          Hello World!
PID : 1
2 processes created
```

2. Model Euclid’s algorithm for Greatest Common Divisor.



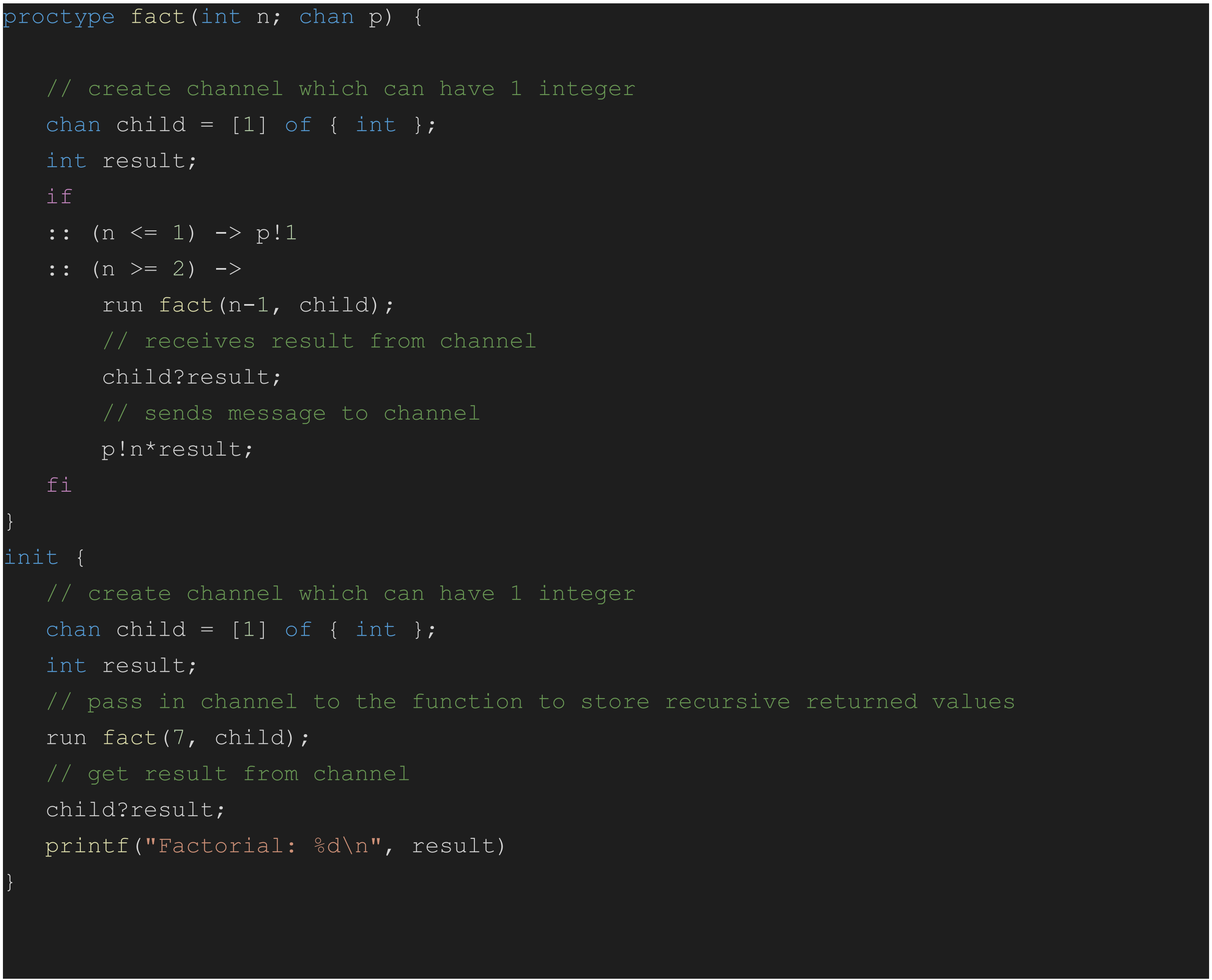
```
proctype euclid_gcd(int a,b) {
   printf("GCD(%d,%d): \n",a,b);
   if
   :: (b == 0) -> printf("%d\n", a);
   :: (b != 0) -> run euclid_gcd(b, a % b);
   fi
}

init {
   run euclid_gcd(21,60);
```

```
}
```

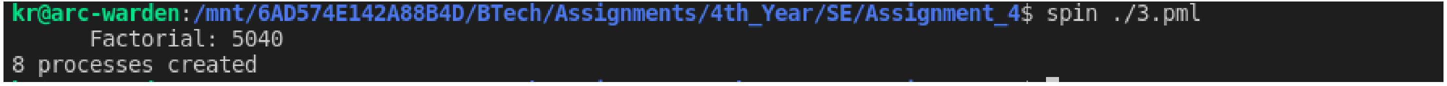


```
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/SE/Assignment_4$ spin ./2.pml
          GCD(21,60):
              GCD(60,21):
                  GCD(21,18):
                      GCD(18,3):
                          GCD(3,0):
                          3
6 processes created
```

3. Create a process factorial(n, c) that recursively computes the factorial of a given non-negative integer “n”.



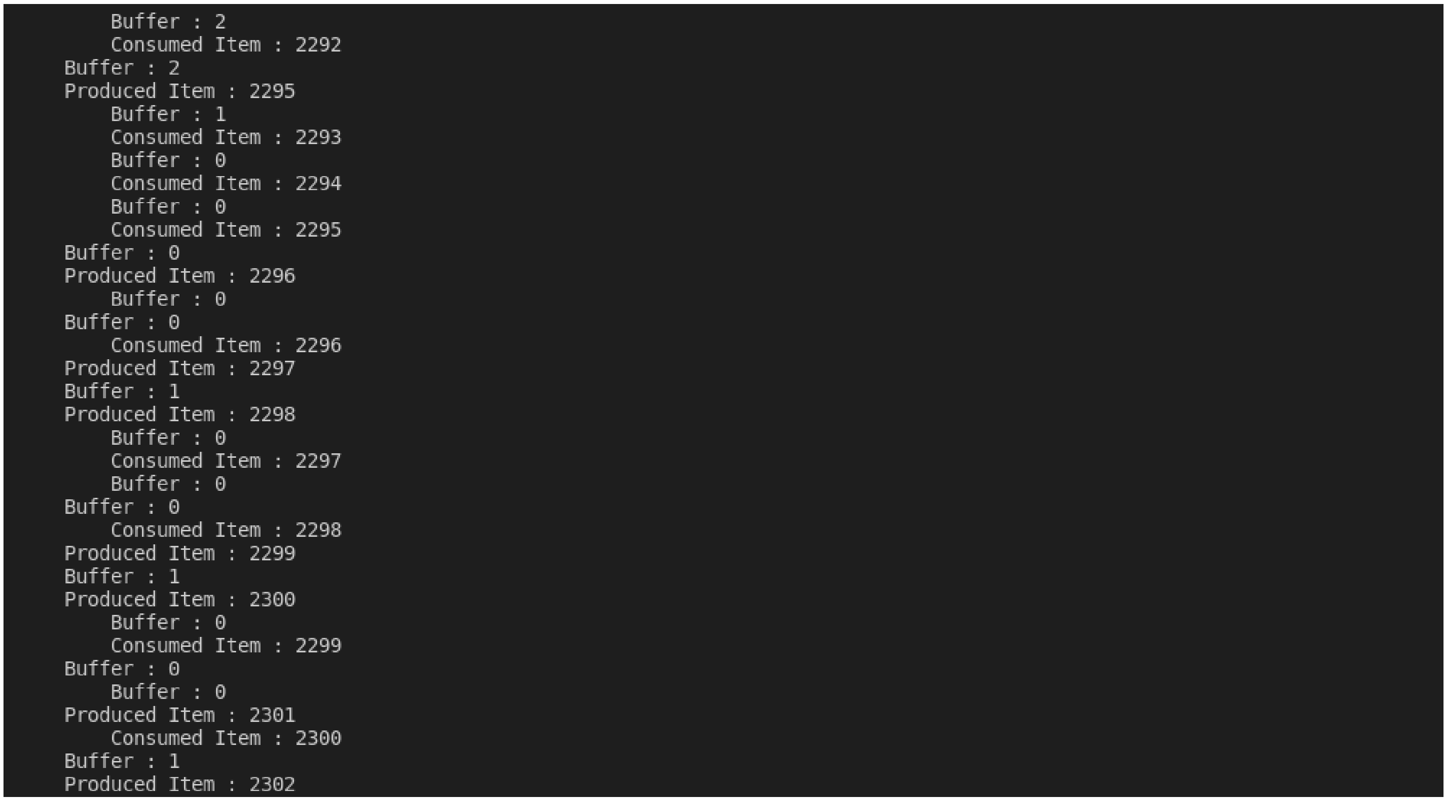
```
proctype fact(int n; chan p) {

   // create channel which can have 1 integer
   chan child = [1] of { int };
   int result;
   if
   :: (n <= 1) -> p!1
   :: (n >= 2) ->
       run fact(n-1, child);
       // receives result from channel
       child?result;
       // sends message to channel
       p!n*result;
   fi
}
init {
   // create channel which can have 1 integer
   chan child = [1] of { int };
   int result;
   // pass in channel to the function to store recursive returned values
   run fact(7, child);
   // get result from channel
   child?result;
   printf("Factorial: %d\n", result)
}
```



```
kr@arc-warden:/mnt/6AD574E142A88B4D/BTech/Assignments/4th_Year/SE/Assignment_4$ spin ./3.pml
      Factorial: 5040
8 processes created
```

4. Create a Promela model for producer-consumer problem with buffer size 5.

```
#define SIZE 5
chan c = [6] of {int};

int buffer = 0;

active proctype producer() {
   int data=0;
   do
   :: buffer < SIZE ->
       buffer = buffer + 1;
       c ! data;
       data++;
       printf("Buffer : %d\n",buffer);
       printf("Produced Item : %d\n",data);
   od
}
active proctype consumer() {
   int data;
   do
   ::  c?data;
       buffer = buffer - 1;
       printf("Buffer : %d\n",buffer);
       printf("Consumed Item : %d\n",data);
   od
}
active proctype monitor() {
   assert (buffer <= SIZE && buffer >= 0);
}
```



```
        Buffer : 2
        Consumed Item : 2292
    Buffer : 2
    Produced Item : 2295
        Buffer : 1
        Consumed Item : 2293
        Buffer : 0
        Consumed Item : 2294
        Buffer : 0
        Consumed Item : 2295
    Buffer : 0
    Produced Item : 2296
        Buffer : 0
    Buffer : 0
        Consumed Item : 2296
    Produced Item : 2297
    Buffer : 1
    Produced Item : 2298
        Buffer : 0
        Consumed Item : 2297
        Buffer : 0
    Buffer : 0
        Consumed Item : 2298
    Produced Item : 2299
    Buffer : 1
    Produced Item : 2300
        Buffer : 0
        Consumed Item : 2299
    Buffer : 0
        Buffer : 0
    Produced Item : 2301
        Consumed Item : 2300
    Buffer : 1
    Produced Item : 2302
```