SYSC 4001 Sample Midterm 1
Duration: 60 minutes
Instructor: T. Kunz

Name: **Student number:**

Question 1. Bakery Algorithm (20 marks)

The *bakery algorithm* provides mutual exclusion to a number of processes, based on the practice observed in bakeries and other shops in which every customer receives a numbered ticket on arrival, allowing each to be served in turn. The algorithm is as follows:

```
Process Pi:
repeat
        choosing[i]:=true;
        number[i]:=max(number[0]..number[n-1])+1;
        choosing[i]:=false;
        for j:=0 to n-1 do {
                while (choosing[j]) {};
                while (number[j]!=0 and (number[j],j)<(number[i],i)){};
        }
        CS
        number[i]:=0;
        RS
forever
```

The arrays *choosing* and *number* are initialized to *false* and 0, respectively. The *i*th element of each array may be read and written by process *i* but only read by other processes. The notation (a,b) < (c,d) is defined as:

(a < c) or (a = c and b < d)

a) Describe the algorithm in words
b) Show that this algorithm avoids deadlocks
c) Show that it enforces mutual exclusion

Question 2. Producer/Consumer Problem with Bounded Buffer (10 marks)

The solution of the bounded buffer problem with finite buffer presented in class allows only at most n-1 entries in the buffer. For your information, the code is included below again:

```
Initialization: S.count:=1;
                N.count:=0;
                E.count:=k;

Producer:       Consumer:       append(v):
Repeat          repeat          b[in]:=v;
  produce v;      wait(N);      in:=(in+1)
  wait(E);        wait(S);           mod k;
  wait(S);        w:=take();
  append(v);      signal(S);    take():
  signal(S);      signal(E);    w:=b[out];
  signal(N);      consume(w);   out:=(out+1)
forever         forever               mod k;
                                return w;
```

a) Explain why this is the case.
b) Modify the algorithm to remedy this deficiency.

Question 3. Banker's Algorithm (10 marks)

a) Consider a system with a total of 150 units of memory, allocated to three processes as shown:

Process	Max	Hold
1	70	45
2	60	40
3	60	15

Apply the banker's algorithm to determine whether it would be safe to grant each of the following requests. If yes, indicate a sequence of terminations that could be guaranteed possible. If no, show the reduction of the resulting allocation table and identify the processes involved in a deadlock:

i) A fourth process arrives, with a maximum memory need of 60 and an initial need of 25 units.

ii) A fourth process arrives, with a maximum memory need of 60 and an initial need of 35 units.

b) Evaluate the banker's algorithm for its usefulness in real life.