Burton Rosenberg

Midterm

October 14, 1993. 5:00–6:15 PM

There are four problems for a total of 100 points. Show all your work, partial credit will be awarded. When there is not enough room on the test page itself, write in the provided blue books and write and sign your name on each one. No notes, no collaboration.

If a problem seems easy: take care!
If a problem seems hard: persevere!

Name: _____

Problem	Credit
1	
2	
3	
4	
Total	

1. Show that these two program fragments are identical. The variables `A` and `B` are declared as `boolean`, and `S1` and `S2` represent two statements.

```
Program Fragment 1:

  if A OR B then
    if A then S1
    else S2 ;

Program Fragment 2:

  if A then S1
    else if B then S2 ;
```

2. Change the following `repeat` loop into an exactly equivalent `while` loop.

```
{Precondition: N is any integer.}
i := 0 ;
repeat
  i := i + 1
until (i*i) > N ;
```

3. Given two lists, L_1 and L_2, the operation of concatenation is defined as third list with all the elements of L_1, in order, followed by all the elements of L_2, in order. For example, if the two lists are:

$$L_1 = a \to b \to c, \qquad L_2 = d \to e \to f$$

then the concatenation would be,

$$\texttt{Concat}(L_1, L_2) = a \to b \to c \to d \to e \to f$$

A list is given by the following code fragment:

```
{    A list is a pointer to a header record       }
{    which contain pointers to the first and last }
{    elements in the list.                        }
type List = ^ListHeader ;
     ListElement = ^ListRecord ;
     ListHeader = record
         first : ListElement ;
         last  : ListElement ;
         end ;
     ListRecord = record
         data : DataType ;
         next : ListElement
         end ;
{    An empty list is a listHeader                }
{    with first = last = nil .                    }
```

Give code for the procedure

```
Procedure Concat(A, B : List ) ;
```

which given two lists `A` and `B`, changes `A` into their concatenation and changes `B` into the empty list. Do this with as efficiently as possible.

4. Improve the speed in the inner loop of the following code fragment.

   (a) As written, how many multiplications are performed as a function of N.

   (b) Give an identically functioning code fragment where only $O(N)$ multiplications are performed.

```
var a : array[1..N,1..N] of integer ;
    i, j : integer ;
begin
  for i := 1 to N do
    for j := i to N do
      a[i,j] := i*i ;
end.
```