Burton Rosenberg
Midterm Answers

1. Show that these two program fragments are identical. The variables $A$ and $B$ are declared as boolean, and $S_1$ and $S_2$ represent two statements.

Program Fragment 1:

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

Program Fragment 2:

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

**Solution:** Transform the first fragment to:

```plaintext
if (A OR B) AND A then S1;
if (A OR B) AND (NOT A) then S2;
```

The absorption identity gives:

$$(A \lor B) \land A = A,$$

which we can apply to the first if. Note: $\lor$ is the symbol for OR, $\land$ is the symbol for AND, and $\neg$ is the symbol for NOT. The law of distribution gives:

$$(A \lor B) \land (\neg A) = (A \land \neg A) \lor (B \land \neg A).$$

The first term of the OR on the right hand side is always false, so it reduces to only the second term. Therefore, we can transform our program again:

```plaintext
if A then S1;
if (B AND NOT A) then S2;
```
which is the same as:

\[
\text{if } A \text{ then } S1 \\
\quad \text{else if } B \text{ then } S2;
\]

which is Program Fragment Two.

2. Change the following \texttt{repeat} loop into an exactly equivalent \texttt{while} loop.

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

\textbf{Solution:} The formula is:

\[\text{repeat } S \text{ until } C \leftrightarrow S; \text{ while } \neg C \text{ do } S.\]

Applying the formula:

\[
i := 0 ; \\
i := i + 1 ; \\
\text{while not}(i*i)>N) \text{ do} \\
i := i + 1 ;
\]

We can neaten this up using simple identities:

\[
i := 1 ; \\
\text{while } (i*i)\leq N \text{ do} \\
i := i + 1 ;
\]

3. Give code for the procedure

\[
\text{Procedure Concat}(A, B : \text{List}) ;
\]
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.

Procedure Concat( \( a, b: \text{list} \) );
begin
  if \( b^.first = \text{nil} \) then begin
    {there is nothing to do in this case}
  end else if \( a^.first = \text{nil} \) then begin
    {\( b \) is not empty, \( a \) is empty.}
    {copy \( b \) to \( a \)}
    \( a^.first := b^.first \);
    \( a^.last := b^.last \);
    \{and make \( b \) empty\}
    \( b^.first := \text{nil} \);  
    \( b^.last := \text{nil} \)
  end else begin
    {both \( a \) and \( b \) are not empty}
    {connect the list together}
    \( a^.last^.next := b^.first \);
    {update list \( a \)}
    \( a^.last := b^.last \);
    \{and make \( b \) empty\}
    \( b^.first := \text{nil} \);  
    \( b^.last := \text{nil} \)
  end
end;

But then we notice that the last three lines of the last two cases are identical, so we can pull them out and put them together:

Procedure Concat( \( a, b : \text{List} \) );
begin
  if \( b^.first \neq \text{nil} \) then begin
    if \( a^.first = \text{nil} \) then \{\( a \) becomes \( b \}\}
      \( a^.first := b^.first \)
    else \{tack on non-empty \( b \) to non-empty \( a \}\}
      \( a^.first^.next := b^.first \);
{update a and make b nil}
a^.last := b^.last ;
b^.first := nil ;
b^.last := nil
end
end ;

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.

```plaintext
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.
```

**Solution:** There are,

$$N + (N - 1) + \ldots + 1 = (N + 1)N/2,$$

multiplications performed.

It would be best to pull the multiplication out of the inner loop, doing it one time for all just before the do loop:

```plaintext
for i := 1 to N do begin
  k := i*i ;
  for j := i to N do
    a[i,j] := k
end ;
```