

Burton Rosenberg

Midterm

OCTOBER 20, 1994.

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 booleans whose values are calculated elsewhere in the program. **S1** and **S2** are two statements.

Program Fragment 1:

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

Program Fragment 2:

```
if (NOT A) then begin
  if (NOT B) then S2
  else if A then S1
end ;
if (B AND A) then S1 ;
```

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. Write code to split a list into two lists by taking elements alternately from the original list. The original list ends up empty. That is, just move pointers, you don't have to new or dispose. Keep the order of the elements and assume no dummy headers.

```

type L_ptr = ^ L_rec ;
  L_rec = record
    i : integer ;
    n : L_ptr ;
  end ;

procedure split_list( var in_list : L_ptr ;
  var out_list_A : L_ptr ;
  var out_list_B : L_ptr ) ;
(* you write the code here *)

```

Here is an example. The symbols such as [x]->[y]->[z] mean list element containing data x pointing to list element containing data y, and so on.

```

Input list:  [1]->[7]->[6]->[3]->[9]->[2]
Output list A:  [1]->[6]->[9]
Output list B:  [7]->[3]->[2]

```

Be careful, the input list can be any length, including zero!

4. Use recursion to compare two lists for equality. Lists are equal when the nodes contain the same data, in the same order. The data types are `L_ptr` and `L_rec`, as in the previous problem. The function header is:

```
function list_compare( a : L_ptr ; b : L_ptr ) : boolean ;  
(* true if a equals b, and false otherwise.
```

```
  Recursion equation to employ:
```

```
  a=b if and only if either
```

```
    (1) they are both empty
```

```
    (2) they are both non-empty,  $a^i = b^i$ 
```

```
        and the lists  $a^n$  and  $b^n$  are equal.
```

```
*)
```

```
  (* you write the code here *)
```