Final review: CPSC 231, spring 2005

Short answer

Short answer question #1:
You are to write a program that will draw a filled square within a 10 x 10 character array. The initial value for each cell of the array is a dot ".". The portions of the array that are a part of the square will be changed to the number sign "#". The program will get the information for the position and size of the square by prompting the user for the row and column (r, c) coordinates for the top left hand corner and for the length of each side. You can assume that the values that are entered by the user won't result in an error condition (e.g., the inner square won't exceed the bounds of the array).

For example if the user types in 2, 2 for the starting row and column coordinate values and 4 for the length, the following square would be drawn:

```
Enter row coordinate of top left corner: 2
Enter column coordinate of top left corner: 2
Enter the length of a side: 4
 .  .  .  .  .  .  .  .  .  .
 .  #  #  #  #  .  .  .  .  .
 .  #  #  #  #  .  .  .  .  .
 .  #  #  #  #  .  .  .  .  .
 .  #  #  #  #  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .
 .  .  .  .  .  .  .  .  .  .
```

Most of the program is already filled in below for this question, you need only add in the missing parts in the "fillInSquare" procedure, which fills in the parts of the array needed to draw the square.

```
program drawSquare (input, output);

const
    SIZE = 10;

type
   CharacterArray = array[1..SIZE,1..SIZE] of char;

procedure initializeSquare (var arr : CharacterArray);
var
   r, c : integer;
begin
   for r := 1 to SIZE do
      for c:= 1 to SIZE do
         arr[r][c] := '.';
end;

procedure getSquareData (var topRow, topColumn, length: integer );
```

```
begin
   write('Enter row coordinate of top left corner: ');
   readln(topRow);
   write('Enter column coordinate of top left corner: ');
   readln(topColumn);
   write('Enter the length of a side: ');
   readln(length);
end;

procedure fillInSquare (    topRow, topColumn, length : integer;
                         var arr                      : CharacterArray);
var
  << Declare your variables here >>




begin
  << Write your answer here >>









  << End of answer space >>
end;

procedure displaySquare (arr : CharacterArray);
var
   r, c : integer;
begin
   for r := 1 to SIZE do
   begin
      for c := 1 to SIZE do
      begin
          write(arr[r][c]);
      end;
      writeln;
   end;
end;

begin
   var topRow, topColumn, length : integer;
   var arr: CharacterArray;

   initializeSquare(arr);
   getSquareData(topRow, topColumn, length);
   fillInSquare(topRow, topColumn, length, arr);
   displaySquare(arr);
end.
```

Short answer question #2
For this question you are to refer to program sa2.p (shown below):

```
program sa2 (output);

const
   LENGTH  = 40;
   NO_BOOKS = 4;
type
   Book = record
             bookTitle  : array [1..LENGTH] of char;
             noPages    : integer;
             authorName : array [1..LENGTH] of char;
          end;

   Node = record
             data : Book;
             next : ^Node;
          end;

BookPointer = ^Node;

procedure add (var myCollection : BookPointer);
var
   i       : integer;
   newNode : BookPointer;
   temp    : BookPointer;
begin
   new(newNode);
   for i := 1 to NO_BOOKS do
   begin
      write('Enter the title of the book: ');
      readln(newNode^.data.bookTitle);
      write('Enter the number of pages for the book: ');
      readln(newNode^.data.noPages);
      write('Enter the author of the book: ');
      readln(newNode^.data.authorName);
      if (myCollection = NIL) then
      begin
         myCollection := newNode;
      end
      else
      begin
         temp := myCollection;
         while (temp^.next <> NIL) do
         begin
            temp := temp^.next;
         end;
         myCollection^.next := newNode;
      end;
      newNode^.next := NIL;
      writeln;
   end;
end;

begin
```

```
   var myCollection : BookPointer;
   myCollection := NIL;
   add(myCollection);
end.
```

While running this program, the user of the program types in the following information for the four books (as prompted to do so in the "add" procedure).

Data for "newNode" the first time that the for loop runs.
Book1
100
Author1

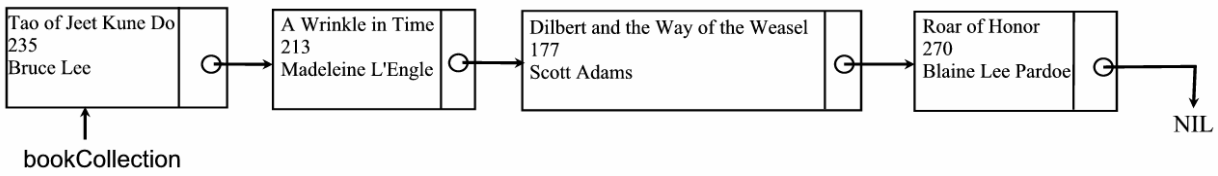
Data for "newNode" the second time that the for loop runs
Book2
100
Author2

Data for "newNode" the third time that the for loop runs
Book3
100
Author3

Data for "newNode" the forth time that the for loop runs
Book4
100
Author4

In the space provided on the next page please draw out the entire linked list in order (including all of the fields of each node) from the first node to the last node. Indicate the end of the list using the NIL pointer.

Example format of the linked list (doesn't match the data above):



<< **Write your answer here** >>:

Multiple choice

1) Assuming that arr is an array of real numbers what does the following procedure do?

```
procedure proc(var arr: RealArray;
                  i, j: integer);
var
   temp : real;
begin
   temp := arr[i];
   arr[i] := arr[j];
   arr[j] := temp;
end;
```

   a) Accept three arguments as input
   b) Swaps two integers
   c) Swaps two array elements
   d) a & b
   e) a & c

2) Please refer to the program below in order to determine which of the following statements are true.

```
program mc1;
var
   num : integer;
begin
   var num: integer;
end.
```

   a) Variable “num” is a global variable.
   b) Variable “num” is local to the main procedure.
   c) There are two variables called “num”, one is global and the other is local.
   d) There is an error in the declaration of variable “num”.
   e) None of the above

For this question please refer to the program shown below.

```
program pointer (output);
begin
   var intPtr : ^integer;
   new(intPtr);
   intPtr^ := 10;
   writeln(intPtr^);
end.
```

3) Which of the following are true of the assignment statement “intPtr^ := 10;”?
   a) It assigns to a pointer the memory address of ten.
   b) It makes the pointer point to memory address ten.
   c) It dereferences a pointer and stores in the dynamically allocated memory the integer ten.
   d) (a) & (b)
   e) None of the above.

JT: Good luck with the real thing!