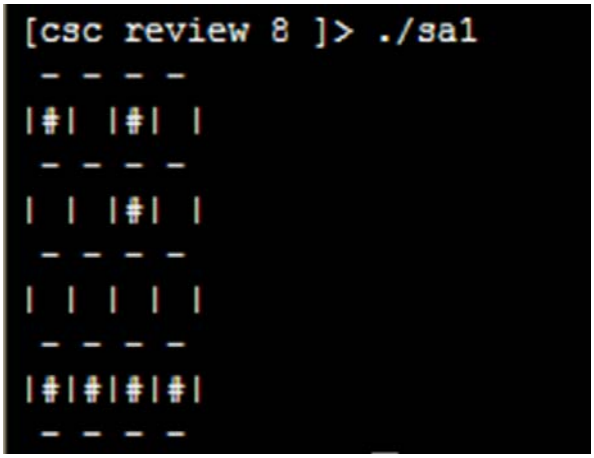
Final review: CPSC 231, fall 2005

Short answer questions:

Short answer question #1:
You are to modify the code for the 'display' procedure below so that when each array element is displayed it appears to be bound: above, below, the left and right side by a line:




```
program unlimitedAdventures (output);

const
   SIZE =  4;

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

procedure initialize (var dungeon :  Level);
begin
   dungeon[1] := '# # ';
   dungeon[2] := '  # ';
   dungeon[3] := '    ';
   dungeon[4] := '####';
end;

procedure display (dungeon : Level);
var
   r : integer;
   c : integer;
begin
   for r := 1 to SIZE do
   begin
      for c := 1 to SIZE do
      begin
         write('|', dungeon[r][c]);
      end;
      writeln('|')
   end;
end;

begin
```

```
   var dungeon : Level;
   initialize(dungeon);
   display(dungeon);
end.
```

<< Write your modified display procedure here (include all necessary code >>

Short answer question #2
In the space provided below you are to show the output of the following program.

```
program list (output);
const
   MAX =  4;
type
   NodePointer = ^ Node;

   Node = record
      data  : integer;
      next : NodePointer;
   end;
procedure initialize (var head :  NodePointer);
var
   i    : integer;
   temp : NodePointer;
begin

   i := 1;
   if (head = NIL) then
   begin
      new(head);
      head^.data := i;
      head^.next := NIL;
   end;
   for i := 2 to MAX do
   begin
      new(temp);
      temp^.data := i;
      temp^.next := head;
      head := temp;
   end;
end;

procedure display (head : NodePointer);
begin
   while (head <> NIL) do
   begin
      write(head^.data, ' ');
      head := head^.next;
   end;
   writeln;
end;


begin
   var head : NodePointer;
   head := NIL;
   initialize(head);
   display(head);
end.
```

<< Write the output to the above program here >>

Short answer question #3:
For this question you are to implement the code for the 'createList' procedure which prompts the user to enter, one at a time, the information for each 'Friend' in the list of friends in an iterative fashion. The loop will stop either when:

1. The maximum capacity for the array has been reached.
2. The user enters a negative value for the phone number. This is a signal that the person has entered the information for the last contact.

```
Entering information for friend number 1
Enter phone number of your friend (negative no. to finish the input of data): 1
Enter the name of your friend: foo1

Entering information for friend number 2
Enter phone number of your friend (negative no. to finish the input of data): 2
Enter the name of your friend: foo2

Entering information for friend number 3
Enter phone number of your friend (negative no. to finish the input of data): 3
Enter the name of your friend: foo3

Entering information for friend number 4
Enter phone number of your friend (negative no. to finish the input of data): 4
Enter the name of your friend: foo4
```

The procedure will take two parameters: 'aFriendList' and 'count'. The first parameter is the array that stores the contact information while the second tracks the number of contacts in the list. Aside for checking that the array isn't overflowed you are not required to do any other extra error checking to receive full credit for this question.

```
program contacts (input, output);

const
   MAX         = 4;
   NAME_LENGTH = 80;

type
   Friend =  record
      phone : integer;
      name  : string[NAME_LENGTH];
   end;

  FriendList = array [1..MAX] of Friend;
```

```
procedure createList (var aFriendList : FriendList;
                      var count       : integer);
```

<< You can declare any local variables here >>

```
begin
```

<< Write your answer here >>

<< End of answer space >>

```
end;
```

```
procedure display (aFriendList : FriendList;
                   count       : integer);
var
   i : integer;
begin
   writeln('DISPLAYING THE LIST OF FRIENDS');
   writeln('------------------------------');
   if (count = 0) then
   begin
      writeln;
      writeln('List is empty, no friends to display');
      writeln;
   end;
   for i := 1 to count do
   begin
      with aFriendList[i] do
      begin
         writeln('Name  : ', name);
         writeln('Phone : ', phone);
         writeln;
      end;
   end;
end;

begin
   var tamFriends : FriendList;
   var count      : integer;

   createList(tamFriends,count);
   display(tamFriends,count);
end.
```

Multiple choice questions:

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

```
procedure proc(var arr : RealArray;
                    i : integer;
                    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 mc2;
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 puts the value 10 in an integer variable called ‘intPtr’.
   b) It makes the pointer point to memory address ten.
   c) It dereferences a pointer and should store in the dynamically allocated memory the integer ten.
   d) (a) & (b)
   e) None of the above.

JT: Good luck with the real thing!