This is a closed note, closed book exam. If you need more space please use the back of the page.

Definition type problem (fill in the blank or matching), here are some good terms to know

1) class 11) pass-by-reference 20) scope
2) private 11) pass-by-value
3) public 12) default
4) double 13) command-line
5) pointer 14) structure
6) linked list 15) NULL
7) array 16) void
8) run-time 17) conditional
9) syntax 18) library
10) pass-by-reference 19) recursive

21) declaration
22) prototype
23) library
24) conditional
25) loop
26) Boolean
27) Class
28) Object
29) structure
**2D arrays** Write a program to create an N by N multiplication table based on ranges that the user enters. The program should both print the table and store it in a two dimensional array. The user should enter the minimum and maximum values for the table. You may assume that the difference between the minimum and maximum values is 10 or less – i.e. the table is no larger than 10 by 10.

For example, if the user enters 7 and 11 the program should print:

```
49 56 63 70 77
56 64 72 80 88
63 72 81 90 99
70 80 90 100 111
77 88 99 110 121
```

(This is only an example, the program should allow the user to enter the values.)

---

**Tracing and/or using pointers and pass-by reference**

Consider the following fragment of code. What is printed?

```cpp
int x = 15;
int y = 10;
int *ptr = &x;
cout << *ptr << endl;
x = y;
cout << *ptr << endl;
*ptr = 7;
cout << y << endl;
y = x;
cout << y << endl;
```

Consider the following fragment of code. What does it print?

```cpp
int function(int a, int &b, int *c){
    a = a + 1;
    b = b + 1;
    *c = *c +1;
    return(a + b + *c);
}

int main(){
    int x = 7;
    int y = 8;
    int z = 9;
    int answer = 0;
```
answer = function(x, y, &z);
cout << x << " " << y << " " << z << " " << answer << endl;
}

**Recursion** Write a recursive function to calculate factorials. A factorial is defined as:

\[ N! = N \times (N-1) \times (N-2) \times \ldots \times 1 \]

Alternatively it can be defined as:

The function should take one positive integer as an argument and return an integer.

**Classes** Consider the following class:

```cpp
class robot{
    public:
        void set_xPosition(int);
        void set_yPosition(int);
        double findDistance(int x, int y);
    private:
        int xPosition;
        int yPosition;
};
```

a) Write the `set_xPosition()` function, which should set the xPosition of the robot, *as long as* the value is in the range 0-50, and otherwise sets the value to 25.

b) Write the `findDistance()` function, which finds and returns the distance between the robot object and the position passed to the function as an argument. The distance, of course, is the square root of the sum of the squares of the x- and y- distances.

c) Write a section of a main() function that creates a robot object *using a pointer*. Then places the robot at location 10,45. Then uses the `findDistance()` function to determine and print the distance from the robot to location 5,5.
Linked lists

Create a node class to store doubles, the class should have an include function for adding nodes, a print function for printing a linked list of nodes, and a find function for finding a particular double in a linked list.

Create and link together 3 nodes.

Given a node class, such as the one from the text or from lab 13, create and link together several nodes. Write a new function to print a linked list backwards.