Lab 11
Function Pointers

Allows your program to determine which function to call dynamically.

Ex:
   double (*func) (double x);
   - This function pointer signature is take a double as parameter, and returns a double.
typedef

typedef stands for type definition. This keyword allows us to form an alias for a datatype!

typedef char box[5][7];

box is now a datatype which refers to a 2D array, with 5 rows and 7 columns, of characters!

box bb, wb, *board[8][8];

bb and wb are both of type box; that is, they are both 2D character arrays with 5 rows and 7 columns! board is a 2D array, with 8 rows and 8 columns, of pointers to box types; that is, each cell of board stores the address of some variable of type box!
for (i = 0; i < 5; ++i) //for each row
    for (j = 0; j < 7; ++j) { //for each column
        wb[i][j] = ' ';
        bb[i][j] = char(219);
    }

Remember that wb and bb are both of type box – that is, a 2D character array with 5 rows and 7 columns. We can treat wb and bb as arrays, because they are arrays!
Filling the board

for (i = 0; i < 8; ++i) //for each row
    for (j = 0; j < 8; ++j)  //for each column
        if ((i+j) % 2 == 0)
            board[i][j] = &wb;
        else
            board[i][j] = &bb;

Remember that pointers store memory addresses. Each element of board is a pointer to a box, and therefore stores the memory address of a box. In the loops above, we alternate between white boxes and black boxes.
How board looks

We only need to make two boxes, \( wb \) and \( bb \). We can then just have each cell in \( board \) point to either \( wb \) or \( bb \)!
Printing the board

for (i = 0; i < 8; ++i) //for each board row
    for (k = 0; k < 5; ++k) { //for each box row
        cout << " " << char(179);
        for (j = 0; j < 8; ++j) //for each board column
            for (l = 0; l < 7; ++l) //for each box column
                cout << (*board[i][j][k][l]); //k is box row, l is box column

        Retrieve box pointer from board, then
dereference. Dereferencing a box pointer
gives us the box being pointed to.

        cout << char(179) << endl;
    }
}
Board and Box Rows

for (i = 0; i < 8; ++i) //for each board row
for (k = 0; k < 5; ++k) //for each box row
Board and Box Columns

for (j = 0; j < 8; ++j) //for each board column
for (l = 0; l < 7; ++l) //for each box column
int main(){
    int i,j,k,l;
    typedef char box[5][7];
    box bb,wb,*board[8][8];
    //fill in bb=black box and wb=white box
    for(i=0;i<5;i++)
        for( j=0;j<7;j++){
            wb[i][j]=' '; 
            bb[i][j]=char(219);
        }
    //fill board with pointers to bb and wb in alternate positions
    for(i=0;i<8;i++)
        for(j=0;j<8;j++)
            if((i+j)%2==0)
                board[i][j]=&wb;
            else
                board[i][j]=&bb;
    // print the board via the pointers in array board
    // first print upper border
    cout<<" ";
    for(i=0;i<7*8;i++)
        cout<<'_' ;
    cout<<endl;
    //now print the board
    for(i=0;i<8;i++)
        for(k=0;k<5;k++){
            for( j=0;j<7;j++){
                cout<<" "<<char(179); //print left border
                for(l=0;l<7;l++)
                    cout<<(*(board[i][j]))[k][l];
                cout<<char(179)<<endl; //at end of line print bar and then newline
            }
        }
    //before exiting print lower border
    cout<<" ";
    for(i=0;i<7*8;i++)
        cout<<char(196);
    cout<<endl;
    return 0;
}
#include <iostream>
using namespace std;

int main(){
    int i,j,k,l;
    char c=' ';
    for( i=0; i<8; i++)
        for(k=0;k<5; k++){
            for(j=0; j<8; j++){
                // print a whole row across
                if((i+j)%2==0) // print a whole row across
                    c=' ';
                else
                    c=char(219);
            for(l=0; l<7;l++)
                cout<<c;
        }
    cout<<endl; // now move to the next line
}
return 0;
}