1. Fill the blank (10 Poins)

```cpp
#include <iostream>
using namespace std;

int main() {
    int q[8][8] = {};
    int r = 0, c = 0;
    q[r][c] = 1;

    nextCol:
    c++;
    if (c == 8) _______________
    r = -1;

    nextRow:
    r++;
    if (_____________) goto backtrack;
    for (int i = 0; i < c; i++)
        if (q[r][i] == 1) goto nextRow;
    for (int i = 1; (r - i) >= 0 && (c - i) >= 0; i++)
        if (q[r - i][c - i] == 1) goto nextRow;
    for (int i = 1; (r + i) < 8 && (c - i) >= 0; i++)
        if (q[r + i][c - i] == 1) goto nextRow;
    q[r][c] = 1;

    _______________

    backtrack:
    _______________
    if (_____________) return 0;
    r = 0;
    while (q[r][c] != 1) r++;
    q[r][c] = 0;
    goto nextRow;

    print:
    static int numSolutions = 0;
    cout << "Solution #" << ++numSolutions << ":\n";
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++)
            cout << q[i][j];
        cout << ":\n";
    }
    goto backtrack;
}
```
2. Write the ok function for 1D 8 Queens without goto (5 Points)

```c
bool ok(int q[], int col) {
    static int mp[3][3] = { { 0,2,1 },
                          { 0,2,1 },
                          { 1,2,0 } };
    static int wp[3][3] = { { 2,1,0 },
                          { 0,1,2 },
                          { 2,0,1 } };
    for (int i=0; i<col;i++) {
        if (______________________) return false;
        // check if the same
        // women married to two different man.
        if((______________< mp[i][q[i]])&&(____________<wp[q[col]][col]) ||
            (___________<mp[col][q[col]])&&(_______________<wp[q[i]][i]))
            return false;
        //check if (Man#i) and (Man#c's wife) both like each other more than
        //their own spouse.
        //Or if (Man#c) and (Man#i's wife) both like each other more than their
        //own spouse.
    }
    return true;
}
```

3. Fill in the blank (10 Points)

```c
bool ok(int q[], int col) {
    static int mp[3][3] = { { 0,2,1 },
                          { 0,2,1 },
                          { 1,2,0 } };
    static int wp[3][3] = { { 2,1,0 },
                          { 0,1,2 },
                          { 2,0,1 } };
    for (int i=0; i<col;i++) {
        if (______________________) return false;
        // check if the same women married to two different man.
        if((______________< mp[i][q[i]])&(&(____________<wp[q[col]][col]) ||
            (___________<mp[col][q[col]])&(&(_______________<wp[q[i]][i]))
            return false;
        //check if (Man#i) and (Man#c's wife) both like each other more than
        //their own spouse.
        //Or if (Man#c) and (Man#i's wife) both like each other more than their
        //own spouse.
    }
    return true;
}
```
4. Give the board layout by using the given helper matrix (5 Points)

```c
int helper[8][7] = {
    {-1},
    {-1},
    {0,1,-1},
    {0,1,-1},
    {0,3,-1},
    {0,2,-1},
    {0,1,2,3,4,5,-1},
    {0,4,5,-1}
};
```

[Extra Points]
Write a function that is given a string and returns an integer. The function looks through the string and finds any INTEGERS. It finds the integers from the string from left to right and concatenated together in that order. Return that integer. (5 Points)
For example: Given “aBk9Hdfmk3am163” the function return 93163