The declaration of Stack as follows:

```
// stack.h -- class declaration for the stack ADT
typedef unsigned long Item;

class Stack
{

private:
    enum {MAX = 10};      // constant specific to class
    Item  * pitems;       // holds stack items
    int size;             // number of elements in stack
    int top;              // index for top stack item
public:
    Stack(int n = MAX);    // creates stack with n elements
    Stack(const Stack & st);
    ~Stack();
    bool isempty() const;
    bool isfull() const;
    // push() returns false if stack already is full, true otherwise
    bool push(const Item & item); // add item to stack
    // pop() returns false if stack already is empty, true otherwise
    bool pop(Item & item);  // pop top into item
    Stack & operator=(const Stack & st);
};
```

Implement all the methods and write a program to demonstrates all the methods, including copy constructor and assignment operator.

2.Create a class Matrix to describe a matrix. The element type is float. One member of the class is a pointer(or a smart pointer) who points to the matrix data.

The two matrices can share the same data through a copy constructor or a copy assignment.

The following code can run smoothly without memory problems.

```
class Matrix{...};

Matrix a(3,4);
Matrix b(3,4);
a(1,2) = 3;
b(2,3) = 4;
Matrix c = a + b;
Matrix d = a;
d = b;
```

```
a is:
0 0 0 0
0 0 3 0
0 0 0 0
b is:
0 0 0 0
0 0 0 0
0 0 0 4
c is:
0 0 0 0
0 0 3 0
0 0 0 4
Before assignment,d is:
0 0 0 0
0 0 3 0
0 0 0 0
After assignment,d is:
0 0 0 0
0 0 0 0
0 0 0 4
```