```
/*  Program: A4P3 - dynarray
    Author: Tom Stutler
    Last Date Modified: 4/8/15

    The intent of this program is to create a class, dynarray, to be
    used in conjunction with a provided main() and myfunc() function.
*/

#include <iostream>

using namespace std;

class dynarray
{
public:
    dynarray() : maxSize(0), usedSize(0)
        { intPtr = new int[maxSize]; }
    dynarray(int sizeParam) : maxSize(sizeParam), usedSize(0)
        { intPtr = new int[maxSize]; }

    void show (int indexParam);
    void set (int indexParam, int valueParam);
    void expand (int sizeParam);

    ~dynarray();

private:
    int maxSize;
    int usedSize;
    int *intPtr;

};

void myfunc();

int main()
{
    int size,more,i;
    dynarray y;

    cout << "Enter dynamic array size: ";
    cin >> size;
    dynarray x(size);

    for(i=0;i<size;i++)
        x.set(i,3*i);

    for(i=0;i<size;i++)
        x.show(i);

    cout << "How much more dynamic array space do you want? ";
    cin >> more;
    x.expand(more);

    for(i=0;i<(size+more);i++)
        x.set(i,5*i);

    for(i=0;i<(size+more);i++)
        x.show(i);

    x.show(size+more+5); //invalid index in show
    x.set(-2,9); //invalid index in set
    y.set(3,6); //empty dynarray set
    y.show(3); //empty dynarray show
    myfunc();

```

```
    return 0;
}

void myfunc()
{
    int i;

    cout << "hi from myfunc...\n";
    dynarray y(5);

    for(i=0;i<5;i++)
        y.set(i,i*i);

    for(i=0;i<5;i++)
        y.show(i);

    cout << "bye from myfunc...\n";
}

void dynarray::show (int indexParam)
{
    if (maxSize==0) {
        cout << "Cannot show - dynarray empty\n";
    } else if (indexParam>usedSize || indexParam<0) {
        cout << "Invalid index in show\n";
    } else {
        cout << *(intPtr+indexParam) << endl;
    }
}

void dynarray::set (int indexParam, int valueParam)
{
    if (maxSize==0) {
        cout << "Cannot set - dynarray empty\n";
    } else if (indexParam>maxSize || indexParam<0) {
        cout << "Invalid index in set\n";
    } else {
        *(intPtr+indexParam) = valueParam;
        usedSize++;
    }
}

void dynarray::expand (int sizeParam)
{
    int *temp = new int[maxSize+sizeParam];

    for (int i=0; i<usedSize; i++) {
        *(temp+i) = *(intPtr+i);
    }

    delete [] intPtr;
    intPtr = temp;
}

dynarray::~dynarray()
{
    cout << "hi from the dynarray destructor...\n";
    delete [] intPtr;
}
```