

```

#include <iostream>
using namespace std;

/**
    2018
*/

class DoubleArray
{
    double *arr;
    double avg = 0;
    int n;
public:
    DoubleArray() {}
    DoubleArray(int nn)
    {
        n = nn;
        arr = new double[n];
    }
    friend istream& operator>>(istream&, DoubleArray&);
    DoubleArray findAvgAndCopy()
    {
        avg = 0;
        for(int i=0; i<n; i++) avg += arr[i];
        avg /= n;

        DoubleArray da(n);
        da.n = n;
        da.avg = avg;
        for(int i=0; i<n; i++) da.arr[i] = arr[i];
        return da;
    }
    void show()
    {
        for(int i=0; i<n; i++)
        {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
    void change()
    {
        for(int i=0; i<n; i++) arr[i] = i+5;
    }

    void max_heap(int m)    /** copied */
    {
        int j, t; t = arr[m]; j = 2 * m;
        while (j < n)
        {
            if (j < n && arr[j+1] > arr[j]) j = j + 1;

```

```

        if (t > arr[j]) break;

        else if (t <= arr[j])
        {
            arr[j / 2] = arr[j];
            j = 2 * j;
        }
    }
    arr[j/2] = t;
    return;
}
void build_maxheap()/** copied */
{
    int k;
    for(k = n/2; k >= 0; k--)
    {
        max_heap(k);
    }
}
};

istream& operator>>(istream& strm, DoubleArray& da)
{
    cout << "Enter items: ";
    for(int i=0; i<da.n; i++) strm >> da.arr[i];
    da.build_maxheap();
}

int main()
{
    DoubleArray da(5);
    cin>>da;
    /** input 1 2 3 4 5 */

    da.show(); /** 5 4 3 2 1 */

    DoubleArray da2;
    da2 = da.findAvgAndCopy();
    da2.show(); /** 5 4 3 2 1 */

    da2.change(); // testing deep copy
    da2.show(); /** 5 6 7 8 9 */

    da.show(); /** 5 4 3 2 1 */
}

```