```
#include<iostream>
#include <cstdlib>
#include<cassert>
using namespace std;
class SA{
private:
        int low, high;
        int* p;
public:

        // default constructor
        // allows for writing things like SA a;

        SA(){low=0; high=-1;p=NULL;}


        // 2 parameter constructor lets us write
        // SA x(10,20);

        SA(int l, int h){
                if((h-l+1)<=0)
                {cout<< "constructor error in bounds definition"<<endl;
                exit(1);}
                low=l;
                high=h;
                p=new int[h-l+1];
        }


        // single parameter constructor lets us
        // create a SA almost like a "standard" one by writing
        // SA x(10); and getting an array x indexed from 0 to 9

        SA(int i){low=0; high=i-1;
        p=new int[i];
        }

        // copy constructor for pass by value and
        // initialization

        SA(const SA & s){
        int size=s.high-s.low+1;
        p=new int[size];
        for(int i=0; i<size; i++)
                p[i]=s.p[i];
        low=s.low;
        high=s.high;
        }

        // destructor

        ~SA(){
                delete [] p;
        }

        //overloaded [] lets us write
        //SA x(10,20); x[15]= 100;

        int& operator[](int i){
                if(i<low || i>high)
                {cout<< "index "<<i<<" out of range"<<endl;
                exit(1);}
                return p[i-low];
        }
        // overloaded assignment lets us assign
        // one SA to another

        SA & operator=(const SA & s){
        if(this==&s)return *this;
        delete [] p;
        int size=s.high-s.low+1;
        p=new int[size];
        for(int i=0; i<size; i++)
                p[i]=s.p[i];
        low=s.low;
        high=s.high;
        return *this;
        }


        // overloads << so we can directly print SAs

        friend ostream& operator<<(ostream& os, SA s);


};

ostream& operator<<(ostream& os, SA s){
int size=s.high-s.low+1;
for(int i=0; i<size; i++)
        cout<<s.p[i]<<endl;
return os;
};


int  main(){
SA a(10), b(3,5);
b[3]=3; b[4]=4; b[5]=5;
int i;
for( i=0;i<10;i++)
a[i]=10-i;

cout<<"printing a the first time“ <<endl;
cout<<a<<endl;


cout<<"printing using []"<<endl;
for( i=0;i<10;i++)
cout<<a[i]<<endl;

// write your own sort
Sort(a,10);

cout<<"printing a the second time“ <<endl;
cout<<a<<endl;

b[4]=12;
cout<<"printing b “ <<endl;
cout<<b<<endl;
a[10]=12; // should print an error message and exit
return 0;

}
```