Data Structures C, D
FAST-NU, Lahore, Fall 2016

Homework 1

Vectors and Iterators

Due on Tuesday August 23 before class | Marked out of 50 points.

This is simple. You have to implement a class Vector (a one dimensional dynamic array), as discussed in class, and an iterator, also as described in class, to traverse the Vector. Additionally, you will add a reverse_iterator and methods to insert and erase elements.

1) The class Vector is defined below; your job is to implement its functions according to our discussion in class and the concepts of programming you acquired in previous courses.

```
template <class T>
class Vector{
    T * arr;
    int size, cap;
public:
    Vector();
    Vector(int);
    //delete existing vector,allocate new one of given capacity, with size=0
    void reallocate(int);
    Vector(const Vector<T>&);// copy constructor
    const Vector<T>& operator=(const Vector<T>&);//assignment operator
    void push_back(const T&);//double capacity when full
    void pop_back();//half capacity when size<cap/2
    int getSize();
    T & operator [] (int)
    ~Vector();//cleap-up memory
};
```

2) Now, as a second step, add an iterator and a reverse_iterator to this class (these are two different classes inside Vector) such that the following two loops work as described in comments.

```
for(Vector<int>::iterator itr=v.begin();
    itr!=v.end();
    itr++){
        //print elements
        cout<<*itr<<' ';
}cout<<endl;

for(Vector<int>::reverse_iterator itr=v.rbegin();
    itr!=v.rend();
    itr++){
        //print elements in reverse
        cout<<*itr<<' ';
}cout<<endl;
```

3) So far we are pretty limited in the use of the vector as the push_back and pop_back functions can only add and remove elements at the end of the array. Now we will add two more member function, **insert** and **erase,** which can add and remove elements from anywhere in the array. The function headers and a sample code are given below. insert and erase should follow the same strategy in resizing the array as push_back and pop_back respectively.

```
insert(Vector<T>::iterator & position, const T & value);
erase(Vector<T>::iterator & position);
```

```
void main(){
    Vector<int> v;//size 256 vector by default
    for(int i=0;i<10;i++){
        v.push_back(i);
    }

    //insert -1 immediately before the number 6
    Vector<int>::iterator itr;
    for(itr=v.begin();*itr!=6;itr++);
    v.insert(itr,-1);

    //erase the number 9
    for(itr=v.begin();*itr!=9;itr++);
    v.erase(itr);

    for(int i=0;i<v.size();i++){
        cout<<v[i]<<' ';
    }cout<<endl;

    //output of this code: 0 1 2 3 4 5 -1 6 7 8
}
```

***

THE END