Q: Templates-Write a program in C++ to implement a generic Stack.

Ans:

Source code:

```
#include<iostream>

#include<string>

#define SIZE 5


using namespace std;


template <class T>

class Stack{

   private:

      int top, capacity;

      T* arr;

   public:

      Stack(int size=SIZE);

      void push(T k);

      T pop();

      T peek();

      int size();

      bool isEmpty();

      bool isFull();

      ~Stack(){

         delete[] arr;

      }
```

```
};
template <class T>
Stack<T>::Stack(int size){
    arr = new T[size];
    capacity = size;
    top = -1;
}

template <class T>
void Stack<T>::push(T k){
    if(isFull()){
        cout<<"Stack is Full!"<<endl;
    }else{
        cout<<"Pushing "<<k<<" to the stack\n";
        arr[++top]=k;
    }
}

template <class T>
T Stack<T>::pop(){
    T top_ele;
    if(isEmpty()){
        cout<<"Stack is Empty!"<<endl;
    }else{
        top_ele = arr[top--];
        cout<<"removing "<<top_ele<<" from Stack\n";
    }
```

```
    return top_ele;
}

template<class T>
T Stack<T>::peek(){
    return arr[top];
}

template<class T>
int Stack<T>::size(){
    return top+1;
}

template<class T>
bool Stack<T>::isEmpty(){
    return top==-1;
}

template<class T>
bool Stack<T>::isFull(){
    return top==capacity-1;
}

int main(){
    Stack<string> st;
    st.pop();
    st.push("Hey");
```

```
    st.push("two");
    cout<<st.pop()<<endl;
    cout<<"size of stack: "<<st.size()<<endl;
    st.push("three");
    st.push("four");
    st.push("five");
    cout<<"size of stack: "<<st.size()<<endl;
    st.push("six");
    if(st.isFull()){
        cout<<"Stack is full\n";
    }
    st.push("seven");
    st.pop();


    return 0;
}
```

Output:

```
Stack is Empty!
Pushing Hey to the stack
Pushing two to the stack
removing two from Stack
two
size of stack: 1
Pushing three to the stack
Pushing four to the stack
Pushing five to the stack
size of stack: 4
Pushing six to the stack
Stack is full
Stack is Full!
removing six from Stack

--------------------------------
Process exited after 0.05603 seconds with return value 0
Press any key to continue . . .
```