

```
1: //Array C++ class
2:
3: #include <iostream>
4: using namespace std;
5:
6: template<class T>
7:
8: class Array
9: {
10: private:
11:     T *A;
12:     int size;
13:     int length;
14:
15: public:
16:     Array()
17:     {
18:         size=10;
19:         A=new T[10];
20:         length=0;
21:     }
22:
23:     Array(int sz)
24:     {
25:         size=sz;
26:         length=0;
27:         A=new T[size];
28:     }
29:
30:     ~Array()
31:     {
32:         delete []A;
33:     }
34:
35:     void Display();
36:     void Insert(int index,T x);
37:     T Delete(int index);
38: };
39:
```

```

40: template<class T>
41: void Array<T>::Display()
42: {
43:     for(int i=0;i<length;i++)
44:         cout<<A[i]<<" ";
45:     cout<<endl;
46: }
47:
48: template<class T>
49: void Array<T>::Insert(int index,T x)
50: {
51:     if(index>=0 && index<=length)
52:     {
53:         for(int i=length-1;i>=index;i--)
54:             A[i+1]=A[i];
55:         A[index]=x;
56:         length++;
57:     }
58: }
59:
60: template<class T>
61: T Array<T>::Delete(int index)
62: {
63:     T x=0;
64:     if(index>=0 && index<length)
65:     {
66:         x=A[index];
67:         for(int i=index;i<length-1;i++)
68:             A[i]=A[i+1];
69:         length--;
70:     }
71:     return x;
72: }
73:
74: int main()
75: {
76:     Array<char> arr(10);
77:
78:     arr.Insert(0,'a');

```

```
79: arr.Insert(1, 'c');
80: arr.Insert(2, 'd');
81: arr.Display();
82: cout<<arr.Delete(0)<<endl;
83: arr.Display();
84: return 0;
85: }
```