

1. Write a Prolog program **sumodd(L,Res)** that computes the sum of integers occurring at the odd positions of L (**assume that L contains at least ONE element**).

E.g. `?- sumodd([1,2,3], Res).`
`Res = 6 //1+2+3`

2. Write a Prolog program **greater(L, X, Res)** that computes all integers in L that are greater than X and stores the result in Res.

E.g. `?- greater([3,6,2,7,4,6,1,9], 4, Res).`
`Res = [6,7,6,9] //6, 7, 6, and 9 in the list are greater than 4`

3. Write a Prolog program **delete_first_k(X,K,L,Res)** that deletes the first k occurrences of X in L and stores the result in Res.

E.g. `?- delete_first_k(1,3,[2,1,3,1,4,1,5,1,6,1,7], Res).`
`Res = [2,3,4,5,1,6,1,7].`

4. Write a Prolog program **prefix(L1,L2)** that checks if L1 is a prefix of L2.

E.g. `?- prefix([1,2,3], [1,2,3,4]).`
`yes.`
`?- prefix([1,2,3], [2,1,2,3]).`
`no.`

5. What's the output of the following Java program

```
class A { public void p() { System.out.println("A.p"); }  
        public void q() { p(); } }  
class B extends A { public void p() { System.out.println("B.p"); } }  
class C extends B { public void q() { p(); } }  
public class Main { public static void main(String args[]) {  
    A r = new B(); r.q(); r = new C(); r.q(); } }
```

6. What's the output of the following C++ program

```
class A { public:  
    virtual void p() { cout << "A::p\n"; }  
    void q() { cout << "A::q\n"; p(); } }  
class B : public A  
{ public: void p() { cout << "B::p\n"; } }  
  
int main() { B b; b.q(); A a; a.q(); a = b; a.q(); }
```