

```
struct Tnode
```

```
{  
    int next[2];  
    int p;  
    int stop;  
    int cnt;  
    int sum;  
};
```

```
// Them x vao multiset
```

```
void InsertNum(int x)
```

```
{  
    int v=0;  
    for(int i=31; i>=0; --i)  
    {  
        int k=(x>>i) & 1;  
        if (Trie[v].next[k]==0)  
        {  
            Trie[v].next[k]=++nT;  
            Trie[nT].p=v;  
        }  
        v=Trie[v].next[k];  
    }  
    ++Trie[v].stop;  
    while (v)  
        ++Trie[v].cnt, Trie[v].sum += x, v=Trie[v].p;  
    ++Trie[v].cnt;  
    Trie[v].sum += x;  
}
```

```
// Bo x khoi multiset
```

```
void RemoveNum(int x)
```

```
{  
    int v=0;  
    for(int i=31; i>=0; --i)  
    {  
        int k=(x>>i) & 1;  
        if (Trie[v].next[k]==0)  
            return;  
        v=Trie[v].next[k];  
    }
```

```

    }
    --Trie[v].stop;
    while (v)
        --Trie[v].cnt, Trie[v].sum -= x, v=Trie[v].p;
    --Trie[v].cnt;
    Trie[v].sum -= x;
}

```

// Tim so dung thu k trong multiset  $k \leq$  so luong so

```

int Rank(int k)
{
    int v=0, x=0;
    for(int i=31; i>=0; --i)
    {
        int C0=(Trie[v].next[0]) ? Trie[Trie[v].next[0]].cnt : 0;
        if (k>C0)
        {
            x |= (1<<i);
            k -= C0;
            v=Trie[v].next[1];
        }
        else
            v=Trie[v].next[0];
    }
    return x;
}

```

// Tinh tong gia tri cac so trong multiset  $\leq x$

```

int SumLE(int x)
{
    int v=0;
    int res=0;
    for(int i=31; i>=0; --i)
    {
        int k=(x>>i) & 1;
        if (k && Trie[v].next[0])
            res += Trie[Trie[v].next[0]].sum;
        if (Trie[v].next[k]==0)
            return res;
        v=Trie[v].next[k];
    }
    res += Trie[v].sum;
    return res;
}

```

```
}
```

```
// Tinh tong cac gia tri tu thu 1 den thu k<=so luong so
```

```
int SumRankLE(int k)
```

```
{
    int v=0;
    int res=0;
    for(int i=31; i>=0; --i)
    {
        int C0=(Trie[v].next[0]) ? Trie[Trie[v].next[0]].cnt : 0;
        int T0=(Trie[v].next[0]) ? Trie[Trie[v].next[0]].sum : 0;
        if (k>C0)
        {
            res += T0;
            k -= C0;
            v=Trie[v].next[1];
        }
        else
            v=Trie[v].next[0];
    }
    res+=Trie[v].sum/Trie[v].stop*k;
    return res;
}
```

```
// Dem xem co bao nhieu so <=x
```

```
int CountLE(int x)
```

```
{
    int v=0;
    int res=0;
    for(int i=31; i>=0; --i)
    {
        int k=(x>>i) & 1;
        if (k && Trie[v].next[0])
            res += Trie[Trie[v].next[0]].cnt;
        if (Trie[v].next[k]==0)
            return res;
        v=Trie[v].next[k];
    }
    res += Trie[v].stop;
    return res;
}
```

