

Abstract (Stack, Priority queue, Dynamic Set, Sequence using queue)

```
import java.io.*;
class kapil
{ public static void main( String args[]) throws Exception
{   DataInputStream u=new DataInputStream(System.in);
    queue t=new queue( );
    String a,b;char p;
    do{a=u.readLine( );
        p=a.charAt(0);
        if (p=='A')
        {   b=a.substring(1);
            t.put(b);
        }
        if (p=='B') t.take( );
        if (p=='C') t.find( );
        if (p=='D') t.pt( );
    }while(1==1);
}
}
class queue
{ private String a[]=new String[10000];
  private int front=0,rear=0;
  public void put(String e)
  { a[rear]=e; rear++;}
  public String take( )
  { front++;return a[front-1]; }
  public void find( )
  { System.out.println(a[front]); }
  public int size( ){ return rear-front; }
  public void pt( )
  { int i;
    for (i=front;i<rear;i++)
        System.out.print(a[i]+",");
    System.out.println( );
  }
}
class stack
{ private queue g;
  public stack( ) { g=new queue( ); }
  public void put(String e){ g.put(e);}
  public String take( )
  { int i;String k;
    for (i=1;i<g.size( );i++)
    { k=g.take( ); g.put(k); }
    return g.take( );
  }
  public void find()
  { int i;String k="";
    for (i=1;i<=g.size();i++)
    { k=g.take();
      g.put(k);
    }
    System.out.println(k);
  }
  public void pt( ) { g.pt( ); }
}
```

- Ax Put x in the system
- B Remove an element
- C Find the top element
- D Print the system

The above program shows the queue behavior.

Aw,At,Ar,Ag,Ak,D,C,B,D,C \Rightarrow wtrgk,w,trgk,t.

When queue t=new queue(); is replaced (in main) by stack t=new stack(); then stack behavior follows.

No array should be used (except in class queue). Only one queue should be used in a class.

Aw,At,Ar,Ag,Ak,D,C,B,D,C \Rightarrow wtrgk,k,wtrg,g.

1. Modify stack. No loop in take/find. Loop in put. At,Ar,Am,Ak,D,C,B,D,C \Rightarrow kmrt,k,mrt,m.
2. Define priority queue. The element taken is the biggest element. No loop in put.
Ag,Ab,Ad,Ak,Af,Ac,Am,Ah,Aj,D,B,D,C,B,D,Ae,D
o/p gbdkfcmhj, gbdkfchj, k, gbdfcjh, gbdfcjhj
3. Modify above only one loop in take. (Hard) *one loop in find*
o/p gbdkfcmhj, bdfgckhj, k, bdfcgjh, bdfcgjhj
4. Modify above. No loop in take/find. *loop in put*
o/p mkjhgfedcb, kjhgfedcb, k, jhgfedcb, jhgfedcb
5. Define dynamic set.
Ax Put x in the system
Bx Remove x from the system
Cx Whether x is present in the system
D Print the system

Make suitable modifications in the main program.

Ap,Ag,Af,At,Ak,D,Cf,D \Rightarrow pgftk,yes,tkpgf.

Ap,Ag,Af,At,Ak,D,Ci,D \Rightarrow pgftk,no,pgftk

Ap,Ag,Af,At,Ak,Bf,DAm,D \Rightarrow tkpg, tkpgm

6. Define sequence.
Axy Put y at location x ($x \leq 9$)
Bx Remove x^{th} element ($x > 9$ permitted)
Cx Find x^{th} element ($x > 9$ permitted)
D Print the sequence
Ex Which element is x

Make suitable modifications in the main program.

A1p,A2g,A3f,A4t,A5k,D,C3,B2,D,C3,D \Rightarrow pgftk,f,ptk,t.

A1p,A2g,A1f,A3t,A3k,D,Ep,D \Rightarrow fpktg 2

7. Define efficient sequence (Hard).

class sequence

```
{ private queue g; private int start;
  public sequence( ){g=new queue( );start=1;}
```

A1p,A2g,A3f,A4t,A5k,D,C3,D \Rightarrow pgftk,f,tkpgf (start=4)

A1p,A2g,A1f,A3t,A3k,D,Ep,D \Rightarrow tgfpk(st=4)2(st=3)ktgfp

Let start=p. It means that the 1st element of the queue is the pth element of the sequence.

3-1 No loop in Put and Find, order Maintain
gbdkfcmhj, gbdkfchj, k, gbdfcjh, gbdfcjhj