```
store_Db_1(Db,n)                    //n is no.of movies
Int i                               //for E movie
rating[n]
For i=0 to n
  read(E[i],rating(i))   //read with space


store_Db_2(Db,n)                    //n is no.of movies
Int i                               //for H movie
rating[n]
For i=0 to n
  read(H[i],rating(i))   //read with space

store_Db_3(Db,n)                    //n is no.of movies
Int i
Rating[n]                           //for M movie
For i=0 to n
  read(M[i],rating(i))   //read with space


list_Db(Db,n)
Int i
For i=0 to n
   print(M[i],rating(i))



popular_rating(Db,n)
max=rating[0]
For i=0 to n
   For j=0 to n
      If M[i]==E[j] and max=rating_[i]>rating[j]
         print(rating[i])
      Else
        print(rating[j])



best_r(E,H,M,n,r)
       merge(E,H,M)
  For i=0 to n
   If M[i]!=r
     Return -1
  For i=0 to n
```

```
rating[i]!=r
b[i]=rating[i]
```

PART 2:

```
heap_sort(a)
Int n                  n is the size
 For i=n/2-1  to 0
    heapify(a,i)
For i=n-1   to  0
    swap(a[0],a[i])
    heapify(a,0)


heapify(a,i)
Int k1=i
Int k2=2*i+1
Int k3=2*i+2
     If k2< n and a[k1]<a[k2])
          k1=k2
     If k3<n  and  a[k1]<a[k3]
           k1=k3
If k1!=i
     swap(a[i],a[k1])
     heapify(a,i)



merge(E,H,M,m,p,r)
Int T,k=0;
  While i<=p  and j<=r
     If E[i] <= E[j]
        T[k]=E[i]
     Else
        T[k]=E[j]


 While i<=p
     T[k]=E[i]
```

While j<=r
T[k]=E[j]