**Main()**

Begin

Read n

Read n strings

Call Compare(Pi,Pj,arr)

Call Insertion-Sort-Lex (string Pi[ ], int Pi.length)

Call Insertion-Sort-Lex (string Pj[ ], int Pj.length)

Call Sum-ASCII (string Pi[])

Call Sum-ASCII (string Pj[])

Return result

if result == True

Call Compare(Pj,Pk)

Call Insertion-Sort-Lex (string Pk[ ], int Pk.length)

Call Sum-ASCII (string Pk[])

else

Call Compare(Pi,Pj)

Call Insertion-Sort-Lex (string Pk[ ], int Pk.length)

Call Sum-ASCII (string Pk[])

Print arr

End

**Compare(Pi,Pj)**

Begin

k=0

If Pi.length<Pj.length

For i=1 to Pi.length

If P[i]>P[j]

k++

For j=1 to P[j].length

Str[j]= P[j]

return

Break

Else if P[i]<P[j]

k++

For j=1 to P[i].length

Str[j]= P[i]

Break

Else

Continue

If Pi.length>Pj.length

If P[i]>P[j]

k++

For j=1 to P[j].length

Str[j]= P[j]

return

Break

Else if P[i]<P[j]

k++

For j=1 to P[i].length

Str[j]= P[i]

Break

Else

Continue

If Pi.length==Pj.length

If P[i]>P[j]

k++

For j=1 to P[j].length

Str[j]= P[j]

return

Break

Else if P[i]<P[j]

k++

For j=1 to P[i].length

Str[j]= P[i]

Break

Else

Continue

if(k==0)

If Pi.length<Pj.length

For j=1 to Pi.length

Str[j]= Pi[j]

Return True

Else if Pi.length>Pj.length

For j=1 to Pj.length

Str[j]= Pj[j]

return False

Else if Pi.length=Pj.length

For j=1 to Pj.length

Str[j]= Pj[j]

For j=Pj.length+1 to 2\*Pj.length

Str[j]= Pj[j]

else

For i=1 to j

If str[i]==Pi[i] && str[i] != Pj[i]

Return True

For i=1 to j

If str[i]==Pj[i] && str[i] != Pj[i]

Return False

End

**Insertion-Sort-Lex (string P[ ], int n)**

Begin

For i=2 to arr.length

temp=arr[i]

j=i-1

while temp<arr[j] && j>=0

arr[j+1]=arr[j]

j=j-1;

arr[j+1]=temp

End

**Sum-ASCII (string str)**

Begin

sum=0;

For i=1 to str.length

sum+=str[i]

return sum

End