Logic Building
Hour Plan - 2

FindString Code

Code:

```
// Write code here...
        int sum=0,sum1=0;
       char c1,c2;
        int i1,i2,i,j;
        String small=new String("abcdefghijklmnopqrstuvwxyz");
        String cap=new String("ABCDEFGHIJKLMNOPQRSTUVWXYZ");
        String s[]=input1.split(" ");
        String res=new String("");
        for(i=0;i<s.length;i++)
        System.out.println(s[i]);
        for(i=0;i<s.length;i++)
        {
        System.out.println(s[i]);
        if(s[i].length()%2==0)
        {

              for(j=0;j<s[i].length()/2;j++)
        {
```

```
c1=s[i].charAt(j);
c2=s[i].charAt(s[i].length()-j-1);
System.out.println(c1+" "+c2);
if(Character.isLowerCase(c1))
i1=small.indexOf(c1)+1;
else
        i1=cap.indexOf(c1)+1;
System.out.println(i1);
if(Character.isLowerCase(c2))
        i2=small.indexOf(c2)+1;
        else
        i2=cap.indexOf(c2)+1;
        System.out.println(i2);

        sum=i1-i2;
        sum1+=Math.abs(sum);



}

}
        else
```

```
{


 for(j=0;j<s[i].length()/2;j++)
{
c1=s[i].charAt(j);
 c2=s[i].charAt(s[i].length()-j-1);
 //System.out.println(c1+" "+c2);
 if(Character.isLowerCase(c1))
i1=small.indexOf(c1)+1;
else
       i1=cap.indexOf(c1)+1;

 if(Character.isLowerCase(c2))
        i2=small.indexOf(c2)+1;
        else
        i2=cap.indexOf(c2)+1;
        System.out.println(i2);

        sum=i1-i2;
        sum1+=Math.abs(sum);
```

```
                }
                char c3=s[i].charAt(s[i].length()/2);
                //System.out.println(c3);
            if(Character.isLowerCase(c3))
                sum1+=small.indexOf(c3)+1;
                else
                    sum1+=cap.indexOf(c3)+1;



        }
System.out.println(sum1);
String s1=String.valueOf(sum1);
res+=s1;
sum1=0;

}
System.out.println(res);
int r=Integer.parseInt(res);
return r;
```

Get Code Through Strings

Code:

```
// Write code here...
                String ar[]=input1.split(" ");
                    int tot=0,len=0;
                for(int i=0;i<ar.length;i++){
                        len+=ar[i].length();
                }
                int sum=0;
                while(len>10){
                        tot=len;
                        sum=0;
                        while(tot>0){

                                        sum+=tot%10;
                                        tot/=10;
                        }
```

```
        len=sum;
    }
    return len;
```

Addition Using Strings

Code:

```
import java.math.BigDecimal;

BigDecimal x = new BigDecimal(input1);
BigDecimal y = new BigDecimal(input2);
return String.valueOf(x.add(y));
```

Simple Encoded Array

Code:

```
//Write code here...
                int sum=input1[input1.length-1];
        for(int i=input1.length-2;i>=0;i--){
                  input1[i]=input1[i]-input1[i+1];
                  sum+=input1[i];
              }
                 Result r=new Result(input1[0],sum);
                 return r;
```

Decreasing Sequence

Code:

```
// Read only region end
	int dcrCount = 0;
		int longestLen = 0;
		int spikeCount = 0;
		boolean flag = false;

		for (int i = 0; i < input2 - 1; i++) {
			if (input1[i] > input1[i + 1]) {
				if (flag == false) {
					flag = true;
					spikeCount++;
				}

				dcrCount++;
				//System.out.println(dcrCount);
				longestLen = dcrCount > longestLen ?
dcrCount : longestLen;
			} else {
				if (flag == true) {
					flag = false;
					dcrCount = 0;
				}
			}
		}
```

```
            if (spikeCount > 0) longestLen++; // fixing fence
post error

            return new Result(spikeCount, longestLen);
```

Most Frequently Occuring Digit

Code:

```
// Write code here...
                int[] ar=new int[10];
                int temp=0,max=0,num=0;
                for(int i=0;i<input1.length;i++){
                        temp=input1[i];
                        while(temp>0){
                                ar[temp%10]+=1;
                                temp=temp/10;
                        }
                }
                for(int j=0;j<ar.length;j++){
                        if(ar[j]>max){
                                max=ar[j];
                                num=j;
                        }
```

```
                    if(ar[j]==max){
                        if(j>num){
                            num=j;
                            max=ar[j];
                        }
                    }
        }
    return num;
```

Sum of Power of Digits

Code:

```
// Write code here...
                Integer sum=0,r=0,prev=0;
                Double f1,f2;
                while(input1>0){
                        r=Integer.valueOf(input1%10);
                        f1=Double.valueOf(r);
                        f2=Double.valueOf(prev);
                        f1=Math.pow(f1,f2);
                        sum+=f1.intValue();
                        prev=Integer.valueOf(r);
                        input1/=10;
                }
        return sum;
```

Sum of Sums of Digits in Cyclic order

Code:

```
// Write code here...
                int last=0,current=0,r=0,sum=0;
                while(input1>0){
                        r=input1%10;
                        current=r+last;
                        input1/=10;
                        sum=sum+current;
                        last=last+r;
                }
        return sum;
```

Identify Possible Words

Code:

```
// Write code here...
                String[] ar=input2.split(":");
                String temp="",fin="";
                int count=0;
                for(int i=0;i<ar.length;i++){
                        temp=ar[i];
                        count=0;
                        if(temp.length()==input1.length()){
                              for(int j=0;j<temp.length();j++){
                                      if(input1.charAt(j)!='_'){

if(Character.toUpperCase(input1.charAt(j))==Character.toUpper
Case(temp.charAt(j))){
                                                      count++;
                                              }
                                      }
```

```
                                    }
                                    if(count==temp.length()-1)
fin=fin+temp.toUpperCase()+":";
                                }
                }
                if(fin=="") return "ERROR-009";
                return fin.substring(0,fin.length()-1);
```

Encoding Three Strings

Code:

```
        //Write code here...
String
f1="",f2="",f3="",m1="",m2="",m3="",l1="",l2="",l3="";
        String out1="",out2="",out3="";
        int d=0;
        //task1
        //input1
        if(input1.length()%3==0){
                d=input1.length()/3;
                f1=input1.substring(0,d);
                m1=input1.substring(d,2*d);
                l1=input1.substring(2*d);
        }
        else if(input1.length()%3==1){
                d=input1.length()/3;
```

```
        f1=input1.substring(0,d);
        m1=input1.substring(d,2*d+1);
        l1=input1.substring((2*d)+1);

}
else{
        d=input1.length()/3;
        f1=input1.substring(0,d+1);
        m1=input1.substring(d+1,2*d+1);
        l1=input1.substring(2*d+1);
}
//input2
if(input2.length()%3==0){
        d=input2.length()/3;
        f2=input2.substring(0,d);
        m2=input2.substring(d,2*d);
        l2=input2.substring(2*d);
}
else if(input2.length()%3==1){
```

```
        d=input2.length()/3;
        f2=input2.substring(0,d);
        m2=input2.substring(d,2*d+1);
        l2=input2.substring((2*d)+1);

}
else{
        d=input2.length()/3;
        f2=input2.substring(0,d+1);
        m2=input2.substring(d+1,2*d+1);
        l2=input2.substring(2*d+1);
}
//input3
if(input3.length()%3==0){
        d=input3.length()/3;
        f3=input3.substring(0,d);
        m3=input3.substring(d,2*d);
        l3=input3.substring(2*d);
}
```

```
else if(input3.length()%3==1){
        d=input3.length()/3;
        f3=input3.substring(0,d);
        m3=input3.substring(d,2*d+1);
        l3=input3.substring((2*d)+1);

}
else{
        d=input3.length()/3;
        f3=input3.substring(0,d+1);
        m3=input3.substring(d+1,2*d+1);
        l3=input3.substring(2*d+1);
}
out1=f1+f2+f3;
out2=m1+m2+m3;
out3=l1+l2+l3;
//task2
String out3_="";
for(int k=0;k<out3.length();k++){
```

```
            if(Character.isUpperCase(out3.charAt(k))){

out3_=out3_+String.valueOf(Character.toLowerCase(out3.char
At(k)));
                }
                else{

out3_=out3_+String.valueOf(Character.toUpperCase(out3.char
At(k)));
                }
        }
        return new Result(out1,out2,out3_);
```