Logic Building Hour Plan - 1

Is Even?

Code:

```
// Write code here...
                if(input1%2==0) return 2;
                else return 1;
```

Is Odd?

Code:

```
if(input1%2!=0) return 2;
                else
                        return 1;
```

Return last digit of the given number

Code:

```
// Read only region end
                if(input1<0)
                        input1=(-1)*input1;


                return input1%10;
```

Return Second last digit of given number

Code:

```
if(input1<0)
                input1=(-1)*input1;
          int c=0;
          int l=Integer.toString(input1).length();
          int r=0;
                if(l==1)
                return -1;
          else
          {
          while(input1>0)
          {
             r=input1%10;
            c++;
          input1/=10;
          if(c==2)
                break;
          }
                return r;
```

Sum of last two digits of two given numbers

Code:

```
if(input1<0)
                    input1=(-1)*input1;
            if(input2<0)
                    input2=(-1)*input2;
            return (input1%10)+(input2%10);
```

Is N an exact multiple of M?

Code:

```
// Write code here...
                    int val=0;
                    if(input1==0 || input2==0) val=3;
                    else if((input1%input2)!=0) val=1;

                    else val=2;

                    return val;
```

Of given 5 numbers,how many are even?

Code:

```
// Write code here

            int cnt=0;

    if(input1<0) input1=(-1)*input1;

            if(input2<0) input2=(-1)*input2;

            if(input3<0) input3=(-1)*input3;

            if(input4<0) input4=(-1)*input4;

            if(input5<0) input5=(-1)*input5;
            if(input1%2==0) cnt++;

            if(input2%2==0) cnt++;

            if(input3%2==0) cnt++;

            if(input4%2==0) cnt++;

            if(input5%2==0) cnt++;


            return cnt;
```

Of given 5 numbers,how many are odd?

Code:

```
// Read only region end
                    int cnt=0;
    if(input1<0) input1=(-1)*input1;
            if(input2<0) input2=(-1)*input2;
            if(input3<0) input3=(-1)*input3;
            if(input4<0) input4=(-1)*input4;
            if(input5<0) input5=(-1)*input5;
            if(input1%2!=0) cnt++;
            if(input2%2!=0) cnt++;
            if(input3%2!=0) cnt++;
            if(input4%2!=0) cnt++;
            if(input5%2!=0) cnt++;
            return cnt;
```

Of given 5 numbers,how many are even or odd?

Code:

```
// Read only region end
                  int cnt=0;
```

```
if(input1<0) input1=(-1)*input1;
        if(input2<0) input2=(-1)*input2;
        if(input3<0) input3=(-1)*input3;
        if(input4<0) input4=(-1)*input4;
        if(input5<0) input5=(-1)*input5;
        if(input1%2!=0) cnt++;
        if(input2%2!=0) cnt++;
        if(input3%2!=0) cnt++;
        if(input4%2!=0) cnt++;
        if(input5%2!=0) cnt++;
        return cnt;
```

Is Prime?

Code:

```
// Read only region end
                    int cnt=0;
                    for(int i=1;i<=input1;i++){
                                if(input1%i==0) cnt++;
                    }
                    if(cnt==2) return 2;
                    else return 1;
```

Factorial of a number

Code:

```
// Read only region end
int i=1;
            int x=1;
            while(i<=input1){
```

```
        x=x*i;
        i++;
}
        return x;
```

Nth Fibonacci

Code:

```
// Read only region end
            int a=0;
            int b=1;
            int c=0;
            int d=3;
            while(d<=input1){
             c=a+b;
```

```
                    a=b;

                    b=c;

                    d++;

            }

            return c;
```

Nth Prime

Code:

```
// Read only region end


        int k=2;

        int d=0,i,c=0;

        int p=0;

        while(d<=input1){
```

```
    for(i=2;i<k/2;i++){
            if(k%i==0){
                    c++;
            }

    }
    if(c==0){
            d++;
            p=k;
    }
    k++;
    c=0;
}
return p;
```

Number of Primes in a Specified range

Code:

```
// Read only region end
        int k=2;
        int d=input1,i,c=0;
        int p=0;
                int cou=0;
        while(d<=input2){
                        for(i=2;i<d;i++){
                                if(d%i==0){
                                                c++;
                                }

                        }

                        if(c==0){
                                cou++;
                System.out.println(d);

            }

            d++;

            c=0;

    }

    return cou;
```

All Digits Count

Code:

```
// Read only region end

            int c=0,r;

            while(input1>0){

            r=input1%10;

                c++;

                input1=input1/10;

            }
```

```
                return c;
```

Unique Digits Count

Code:

```
        // Read only region end
                int c=0,r,i;
                        int h[]=new int[10];
                        while(input1>0){
                        r=input1%10;
                                h[r]++;
                                input1=input1/10;
                        }
                        for(i=0;i<10;i++){
                                if(h[i]>0){
```

```
                            c++;
                    }
            }
            return c;
```

Non - Repeated Digits Count

Code:

```
// Read only region end

        int c=0,r,i;
            int h[]=new int[10];
            while(input1>0){
            r=input1%10;
                    h[r]++;
            input1=input1/10;
        }
        for(i=0;i<10;i++){
            if(h[i]==1){
                c++;
            }
        }
        return c;
```

digitSum : Sum of all digits in N

Code:

```
// Read only region end
        boolean b=true;
        int r,sum=0;
        int x=input1,res=0;
        input1=Math.abs(input1);
        while(b){
```

```
        while(input1>0){
                r=input1%10;
                sum=sum+r;
                input1=input1/10;
        }
        if(sum<10){
                b=false;
        }
        else{
                input1=sum;
                sum=0;
        }
    }
    if(x<0){
        res=-sum;
    }
    else{
        res=sum;
    }
```

```
                return res;
```

digitSum even : Sum of even digits in N.

Code:

```
// Read only region end

                int r,sum=0;
                        while(input1>0){
                                r=input1%10;
                                if(r%2==0){
                                sum=sum+r;
                                }
                                input1=input1/10;
```

```
          }

      return sum;
```

digitSum odd : sum of odd digits in N

Code:

```
// Read only region end

      int r,sum=0;
          while(input1>0){
              r=input1%10;
              if(r%2==1){
              sum=sum+r;
              }
```

```
                                        input1=input1/10;
                              }

                    return sum;
```

digitSum opt : Sum of even or odd digits

Code:

```
// Read only region end
     // Write code here...
     if(input2.equals("odd"))
     {
          int sum=0;
     while(input1>0)
          {
          int r=input1%10;
```

```
    if(r%2==1)
    {
    sum+=r;
    }
    input1/=10;



}
return sum;
}
else
{
    int sum=0;
while(input1>0)
    {
    int r=input1%10;
```

```
        if(r%2==0)
        {
        sum+=r;
        }
        input1/=10;



    }
     return sum;
```

Is Palindrome Number?

Code:

```
 // Read only region end
    // Write code here
    int temp=input1;
    int rev=0;
    while(input1>0)
    {
    rev=rev*10+input1%10;
     input1/=10;
    }
    if(rev!=temp)
    return 1;
    return 2;
```

Is Palindrome Possible?

Code:

```
int h1[] = new int[26];
        for(int i = 0; i < h1.length; i++) {
```

```
h1[i] = 0;
    }
    System.out.println(h1);
    int i;
  while(input1>0)
  {
     h1[input1%10]++;
     input1 /= 10;

  }



  int odd=0;
  for(i=0;i<10;i++)
  {
        if(h1[i] == 1)
          odd++;
     if(odd>1)
          return 1;
  }
  return 2;
```

Create PIN using alpha, beta, gamma

Code:

```
int u1=input1%10,u2=input2%10,u3=input3%10;
int t1=(input1/10)%10,t2=(input2/10)%10,t3=(input3/10)%10;
int h1=input1/100,h2=input2/100,h3=input3/100;
int u=Math.min(u1,Math.min(u2,u3));
int t=Math.min(t1,Math.min(t2,t3));
int h=Math.min(h1,Math.min(h2,h3));
int th=Math.max(u1,Math.max(u2,Math.max(u3,Math.max(t1,Math.max(t2,Math.max(t3,Math.max(h1,Math.max(h2,h3))))))));
int num=th*1000+h*100+t*10+u;
```

```
             return num;
```

Weight of a hill pattern

CODE :

```
       int sum=0,i,j;
    for(i=0;i<input1;i++)
    {

        for(j=0;j<=i;j++)
            sum+=input2;
        input2=input2+input3;
        //weight=input2+input3;
    }



    return sum;
```

Return second word in Uppercase

Code:

```
  String s[]=input1.split(" ");
   if(s.length==1)
       return "LESS";
   String s1=s[1];
   s1=s1.toUpperCase();
   return s1;
```

is Palindrome (string)

Code:

```
   input1=input1.toLowerCase();
   int i,flag=1;
   for(i=0;i<input1.length()/2;i++)
   {
```

```
if(input1.charAt(i)!=input1.charAt(input1.length()-i-1)
)
        {
            flag=0;
            break;
        }
    }
    if(flag==0)
        return 1;
    return 2;
```

weight of string

Code:

```
    String small="abcdefghijklmnopqrstuvwxyz";
    int sum=0,i;
```

```
for(i=0;i<input1.length();i++)

{
if(input2==0)
{
    char c=input1.charAt(i);
    if(Character.isUpperCase(c))
        c=Character.toLowerCase(c);
     if(c!='a'&&c!='e'&&c!='i'&&c!='o'&&c!='u')
    {
    int index=small.indexOf(c);
    if(index>=0)
        sum+=index+1;
    }
    else
        sum+=0;
    }
        else
        {
            char c=input1.charAt(i);
        if(Character.isUpperCase(c))
            c=Character.toLowerCase(c);
            int index=small.indexOf(c);
        if(index>=0)
            sum+=index+1;
            else
            sum+=0;
        }
}
    return sum;
```

Most Frequent Digit

Code

```
int h[] = new int[10];
      for(int i = 0; i < h.length; i++) {
     h[i] = 0;
      }
  int i;


if(input1==0&&input2==0&&input3==0&&input4==0
)
     return 0;
  if(input1==0)
     h[0]++;
  if(input2==0)
     h[0]++;
  if(input3==0)
```

```
    h[0]++;
if(input4==0)
    h[0]++;
while(input1>0)
{
    h[input1%10]++;
    input1/=10;
}
while(input2>0)
{
    h[input2%10]++;
    input2/=10;
}
while(input3>0)
{
    h[input3%10]++;
    input3/=10;
}
while(input4>0)
{
    h[input4%10]++;
    input4/=10;
}
int index=0,max=-1;
for(i=0;i<10;i++)
{
    if(max<=h[i])
    {
        max=h[i];
        index=i;
    }
}
```

```
return index;
```