Logic Building Hour Plan -3

Generate series and find nth element

Code:

```
// Write code here...
        int i=3,diff=0,next=0;
        while(i<input4){
            diff=input2-input1;
            next=input3+diff;
            input1=input2;
            input2=input3;
            input3=next;
            i++;
        }
        return next;
```

Find Result after alternate add-sub on N

Code:

```
//Write code here
            int glob=0;

       if(input2==1){

                  for(int i=0;i<=input1;i++){
                        if(i%2==0){
                              glob=glob+(input1-i);
                        }
                        else glob=glob-(input1-i);
                  }
            }
            else{
                  for(int i=0;i<=input1;i++){
                        if(i%2==0 && i!=0){
                              glob=glob-(input1-i);
                        }
                        else glob=glob+(input1-i);
                  }
            }
      return glob;
```

Find Password(stable - unstable)

Code:

```
// Read only region end
            int sumOfStable = 0;
            int sumOfUnstable = 0;

            if (isStable(input1)) sumOfStable += input1;
            else sumOfUnstable += input1;

            if (isStable(input2)) sumOfStable += input2;
            else sumOfUnstable += input2;

            if (isStable(input3)) sumOfStable += input3;
            else sumOfUnstable += input3;

            if (isStable(input4)) sumOfStable += input4;
            else sumOfUnstable += input4;

            if (isStable(input5)) sumOfStable += input5;
            else sumOfUnstable += input5;

            System.out.println(sumOfStable + " :: " +
sumOfUnstable);
```

```
            System.out.println("isStable: " + isStable(input1) +
isStable(input2) + isStable(input3) + isStable(input4) +
isStable(input5));

            return sumOfStable - sumOfUnstable;
      }

      public static boolean isStable(int num) {
            boolean isStable = true;
            int[] freq = new int[10];
            String numStr = String.valueOf(num);

            for (int i = 0; i < numStr.length(); i++) {

freq[Integer.parseInt(String.valueOf(numStr.charAt(i)))]++;
            }

            System.out.println(Arrays.toString(freq));

            int firstFreq = 0;
            for (int i = 0; i < 10; i++) {
                  if (freq[i] > 0) {
                        firstFreq = freq[i];
                        break;
                  }
            }
```

```
System.out.println("firstFreq: " + firstFreq);

for (int i = 0; i < 10; i++) {
        if (freq[i] != 0 && freq[i] != firstFreq) {
                isStable = false;
                break;
        }
}
System.out.println("isStable: " + isStable);

return isStable;
```

Calculate Sum of non-Prime index values

Code:

```
// Read only region end
        // Write code here...
        int sum=input1[0]+input1[1];
        int i,j,flag;
        for(i=3;i<input2;i++)
        {
            flag=1;
            for(j=2;j<=Math.sqrt(i);j++)
            {
                if(i%j==0)
                {
                    flag=0;
                    break;
                }
            }
            System.out.println(flag);
            if(flag==0)
                sum+=input1[i];
        }
        return sum;
```

Find the one digit to be removed from the palindrome

Code:

```
// Write code here...
int[] h=new int[10];
int t=input1;
int r,rev=0;
while(input1>0)
{
    r=input1%10;
    rev=rev*10+r;
    input1/=10;
}
if(rev==t)
    return -1;
input1=t;
while(input1>0)
{
    h[input1%10]++;
    input1/=10;
}
//String s=String.valueOf(input1);

int index=-1,i;
```

```
for(i=0;i<10;i++)
{
    if(h[i]%2==1)
    {
    index=i;
    }
}
System.out.print(index);
return index;
```

The Nambair Number Generator

Code:

```
// Read only region end
            String s=input1;
            int len=s.length();
            int a[]=new int[len];
            for(int i=0 ;i <len ;i++)
            {
                  a[i]=(s.charAt(i)-'0');
            }
            System.out.println(Arrays.toString(a));
            int i=0;
            String temp="";
            int k=a[i];
            int evenflag,oddflag;
            if(k%2==0)
            {
                  evenflag=1;
                  oddflag=0;
            }
            else
            {
                  evenflag=0;
                  oddflag=1;
            }
```

```
while(i<len)
{
        if(i==len-1)
        {
                System.out.print(k);
                temp+=k;
                break;
        }
        if((k%2!=0)&&(oddflag==1))
        {
                k+=a[i+1];
                i++;
        }
  else if((k%2==0)&&(evenflag==1))
        {
                k+=a[i+1];
                i++;
        }
        else
        {
                System.out.print(k+" ");
                temp+=k;
                i=i+1;
                k=a[i];
                if(k%2==0)
           {
                evenflag=1;
                  oddflag=0;
          }
else
      {
       evenflag=0;
       oddflag=1;
      }
       }

}
return Integer.parseInt(temp);
```

User Id Generation

Code:

```
// Read only region end
        String firstName = input1;
                String lastName = input2;
                int pin = input3;
                int N = input4;

                String longerName;
                String smallerName;
                StringBuilder userId = new StringBuilder();

                if (firstName.length() > lastName.length()) {
                        longerName = firstName;
                        smallerName = lastName;
                } else if (firstName.length() < lastName.length())  {
                        longerName = lastName;
                        smallerName = firstName;
                } else {
                        if (firstName.compareTo(lastName) < 1 ) {
                                longerName = lastName;
                            smallerName = firstName;
                        } else {
                                longerName = firstName;
```

```
                    smallerName = lastName;
                }
            }


userId.append(smallerName.charAt(smallerName.length() -
1));
            userId.append(longerName);

            for (int i = 0; i < userId.length(); i++) {
                if (Character.isUpperCase(userId.charAt(i)))
                    userId.setCharAt(i,
Character.toLowerCase(userId.charAt(i)));
                else
                    userId.setCharAt(i,
Character.toUpperCase(userId.charAt(i)));
            }

            userId.append(String.valueOf(pin).charAt(N - 1));

userId.append(String.valueOf(pin).charAt(String.valueOf(pin).l
ength() - N));

            return userId.toString();
```

Message Controlled Robot Movement

Code:

```
// Read only region end
            int X = input1;
            int Y = input2;
            String currentPos = input3;
            String msg = input4;

            int currX = Integer.parseInt(currentPos.split("-")[0]);
            int currY = Integer.parseInt(currentPos.split("-")[1]);
            String currD = currentPos.split("-")[2]; // E/W/N/S
            String[] instructions = msg.split(" "); // M L R M M L M
...
            StringBuilder output = new StringBuilder();

            System.out.println(Arrays.toString(instructions));
            System.out.println("Curr: " + currX + currY + currD);

            for (int i = 0; i < instructions.length; i++) {
                  System.out.print(instructions[i] + ":: ");
                  if (instructions[i].equals("M")) {
                        if (currD.equals("E") && (currX + 1 > X )) {
                              output.append("-ER");
                              break;
```

```
                    }
                    if (currD.equals("W") && (currX - 1 < 0 )) {
                        output.append("-ER");
                        break;
                    }
                    if (currD.equals("N") && (currY + 1 > Y )) {
                        output.append("-ER");
                        break;
                    }
                    if (currD.equals("S") && (currY - 1 < 0 )) {
                        output.append("-ER");
                        break;
                    }

                    if (currD.equals("E")) currX++;
                    else if (currD.equals("W")) currX--;
                    else if (currD.equals("N")) currY++;
                    else if (currD.equals("S")) currY--;
                } else {
                    if (currD.equals("E") &&
instructions[i].equals("L"))
                      currD = "N";
                    else if (currD.equals("E") &&
instructions[i].equals("R"))
                        currD = "S";
                        else if (currD.equals("W") &&
instructions[i].equals("L"))
                              currD = "S";
                        else if (currD.equals("W") &&
instructions[i].equals("R"))
                              currD = "N";
                        else if (currD.equals("N") &&
instructions[i].equals("L"))
                              currD = "W";
                        else if (currD.equals("N") &&
instructions[i].equals("R"))
                              currD = "E";
                        else if (currD.equals("S") &&
instructions[i].equals("L"))
                              currD = "E";
                        else if (currD.equals("S") &&
instructions[i].equals("R"))
                              currD = "W";
                  }

                  output.delete(0, output.length());
                  output.append(currX + "-" + currY + "-" + currD);
                  System.out.println(output);
            }

            return output.toString();
```