CSE3032 - Competitive Programming
WIN SEM (2022-2023) AMR
Class Number: AP2022236001007
Slot: L11+L12+L19+L20
ASSIGNMENT - 2

Last Date for Submission: Wednesday (28-01-2023)

Name: Chilamakuru Sai Venkata Yaswanth

Reg No: 20BCI7076

Write the program using (C / C++ / Java / Python) to print the following patterns.

S.No	Problem Name	Link
1	Recursive Digit Sum	https://www.hackerrank.com/challenges/recursive-digitsum/problem?isFullScreen=true
2	Power Sum	https://www.hackerrank.com/challenges/the-powersum/problem?isFullScreen=true
3	Recursive Function	https://www.hackerearth.com/practice/basicprogramming/implementation/basics-ofimplementation/practice-problems/algorithm/recursivefunction/
4	PRIME1 - Prime Generator	https://www.spoj.com/problems/PRIME1/
5	PALIN - The Next Palindrome	https://www.spoj.com/problems/PALIN/
6	GCD Problem	https://www.hackerearth.com/problem/algorithm/gcd-witha-twist/
7	The GCD function	https://www.hackerearth.com/problem/algorithm/gcdfunction-9fe49c14/
8	GCD LCM	https://www.hackerearth.com/problem/algorithm/gcd-lcm1-37ac1ce5/
9	Fibonacci and GCD	https://www.hackerearth.com/problem/algorithm/fibonacciand-gcd/
10	Hardik and GCD	https://www.hackerearth.com/problem/algorithm/temp/

Note:

- If Code similarity is found, assignment will not be considered and Zero (0) Marks will be awarded.
- You have to upload a single document consisting of all the above programs and corresponding Output.
- You will be asked to explain the code, run and show the same program in the respective platforms (hacker rank / hacker earth / spoj)

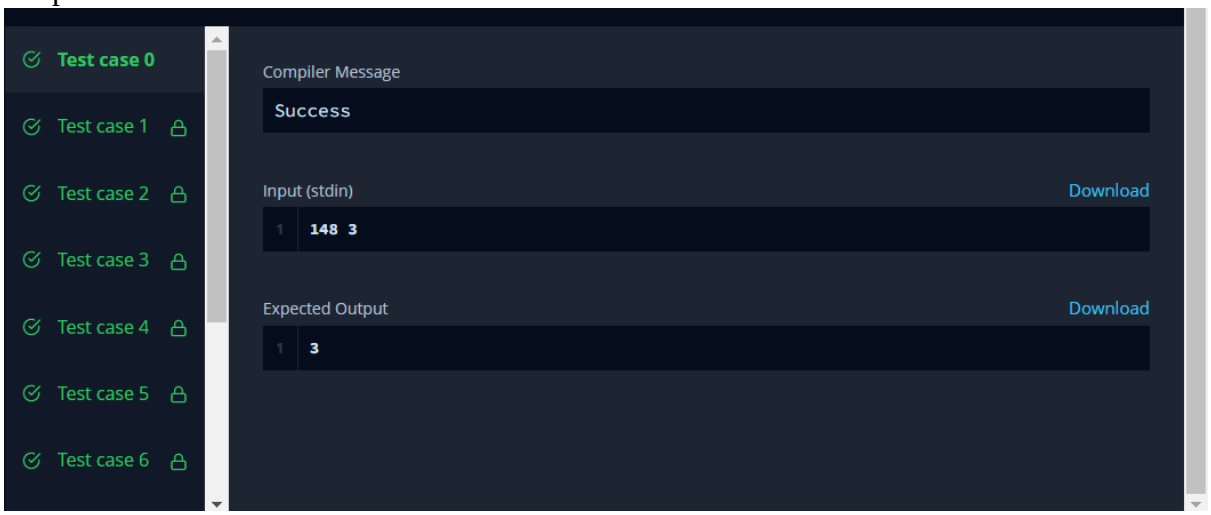
1) Recursive Digit Sum:

Code:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;
public class Main{
    static int DigitSum(String n, int k){
        long a=n.length()*k;
        int c=0;
        while(a>1l){
            long p=0l;
            for(int x=0;x<n.length();x++){
                p=p+Long.parseLong(""+n.charAt(x));
            }
                if(c==0){
                    p=p*k;
                    c++;
                }
                    n=Long.toString(p);
                    a=n.length();
        }
        return Integer.parseInt(n);
}
private static final Scanner scanner=new Scanner(System.in);
public static void main(String [] args) throws IOException{
    BufferedWriter bufferedWriter=new BufferedWriter(new FileWriter(Sys
tem.getenv("OUTPUT_PATH")));
    String [] nk=scanner.nextLine().split(" ");
    String n=nk[0];
    int k=Integer.parseInt(nk[1]);
    int result=DigitSum(n, k);
```

```
    bufferedWriter.write(String.valueOf(result));
    bufferedWriter.newLine();
    bufferedWriter.close();
    scanner.close();
    }
}
```

Output:



2) Power Sum:

Code:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;
class Main{
    static boolean [][] powerSum;
    static ArrayList<Integer> total=new ArrayList<>();
    public static void ps(int [] arr,int i,int sum,ArrayList<Integer> p
){
        if(i==0 && sum!=0 && powerSum[0][sum]){
            p.add(arr[i]);
            total.add(p.size());
            p.clear();
            return;
        }
        if(i==0 && sum==0){
            total.add(p.size());
            p.clear();
            return;
```

```
        }
        if(powerSum[i-1][sum]){
            ArrayList<Integer> b=new ArrayList<>();
            b.addAll(p);
            ps(arr,i-1,sum,b);
        }
        if (sum>=arr[i] && powerSum[i-1][sum-arr[i]]){
            p.add(arr[i]);
            ps(arr,i-1,sum-arr[i], p);
        }
    }
    public static int powerSum(int X,int N){
        int count=0;
        if(X==0){
            return 0;
        }
        else{
            for(int i=1;Math.pow(i,N)<=X;i++){
                count++;
            }
            int arr[]=new int[count];
            for(int i=0;i<count;i++){
                arr[i]=(int)Math.pow(i+1,N);
            }
            if(count<=1||X<=0){
                return 0;
            }
            powerSum=new boolean[count][X+1];
            for (int i=0;i<count;++i){
                powerSum[i][0]=true;
            }
            if (arr[0]<=X)
            powerSum[0][arr[0]]=true;
            for (int i=1;i<count;++i)
            for (int j=0;j<X+1;++j)
            powerSum[i][j]=(arr[i]<=j) ? (powerSum[i-
1][j] ||powerSum[i-1][j-arr[i]]): powerSum[i-1][j];
            ArrayList<Integer> p=new ArrayList<>();
            ps(arr,count-1,X,p);
            return total.size();
        }
    }
}
public class Solution{
    public static void main(String [] args) throws IOException{
```

```
        BufferedReader bufferedReader=new BufferedReader(new InputStrea
mReader(System.in));
        BufferedWriter bufferedWriter=new BufferedWriter(new FileWriter
(System.getenv("OUTPUT_PATH")));
        int X=Integer.parseInt(bufferedReader.readLine().trim());
        int N=Integer.parseInt(bufferedReader.readLine().trim());
        int result=Main.powerSum(X,N);
        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();
        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

Output:

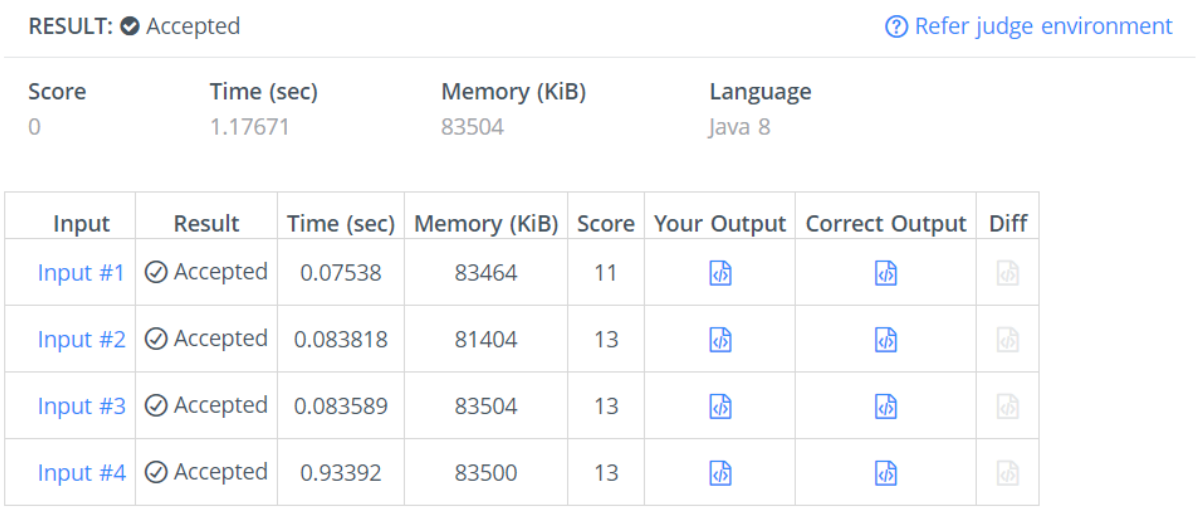
3) Recursive Function:

Code:

```
import java.util.*;
public class Main{
    public static long f(long x,long y){
        if(x==0 && y>=0){
            return (y+1)%1000;
        }
        else if(x>0 && y==0){
            return f(x-1,1);
        }
        else if(x>0 && y>0){
            long a=f(x,y-1);
            return f(x-1,a);
        }
        return 0l;
    }
    public static void main(String [] args) throws Exception{
        Scanner sc=new Scanner(System.in);
```

```
        String name=sc.nextLine();
        String [] parts=name.split(" ");
        long x,y;
        x=Long.parseLong(parts[0]);
        y=Long.parseLong(parts[1]);
        long result=f(x,y);
        String priceStr=Long.toString(result);
        if(priceStr.length()<4){
            priceStr="0000"+priceStr;
        }
        int last3Index=priceStr.length()-3;
        int last3Digits=Integer.parseInt(priceStr.substring(last3Index));
        String padded=String.format("%03d",last3Digits);
        System.out.println(padded);
    }
}
```

Output:



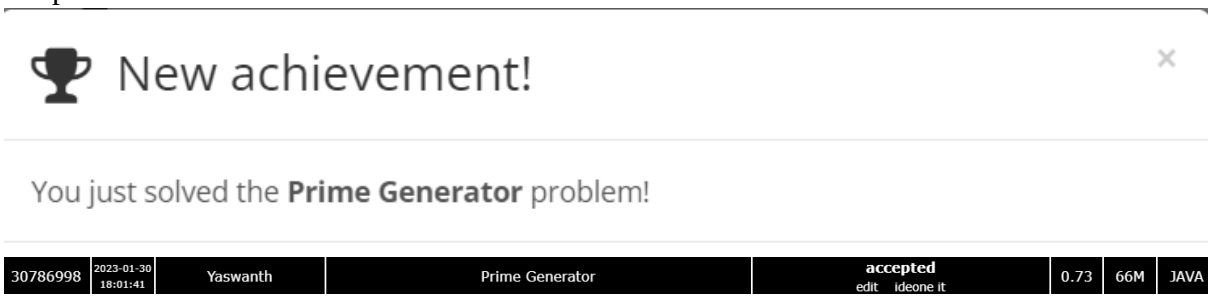
4) Prime1-Prime Generator:

Code:

```
import java.io.*;
import java.util.*;
import java.lang.Math.*;
public class Main{
        public static void main(String [] args){
                Scanner sc=new Scanner(System.in);
                int [] prime=new int[6000];
                int num=0;
                prime[num++]=2;
                for(int i=3;i<=32000;i+=2){
```

```
                boolean isprime=true;
                double c=Math.sqrt(i)+1.0;
                for(int j=0;j<num;j++){
                        if (j>=c)
                        break;
                        if(i%prime[j]==0){
                                isprime=false;
                                break;
                        }
                }
                if(isprime)
                prime[num++]=i;
        }
        int T,N,M;
        T=sc.nextInt();
        for(int t=0;t<T;t++){
                if(t>0)
                System.out.println("");
                M=sc.nextInt();
                N=sc.nextInt();
                if(M<2)
                M=2;
                boolean [] isprime=new boolean[100001];
                for(int j=0;j<100001;j++){
                        isprime[j]=true;
                }
                for(int i=0;i<num;i++){
                        int p=prime[i];
                        int start;
                        if (p>=M)
                        start=p*2;
                        else
                        start=M+((p-M%p)%p);
                        for (int j=start;j<=N;j+=p){
                                isprime[j-M]=false;
                        }
                }
                for(int i=M;i<=N;i++){
                        if(isprime[i-M])
                        System.out.println(i);
                }
        }
    }
}
```

Output:



5) Palin-The Next Palindrome:

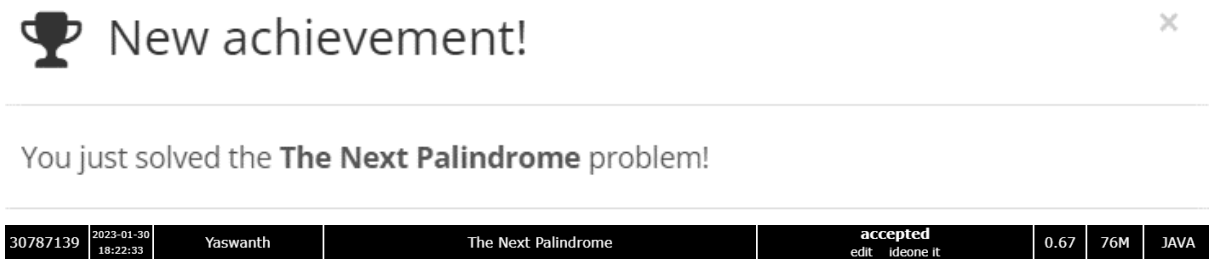
Code:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.*;
public class Main {
	public static char [] getNextNum(char [] a){
		boolean carry = true;
		for(int i=a.length-1;i>=0;i--){
			if(!carry)
			break;
			if(a[i]=='9'){
				carry=true;
				a[i]='0';
			}
			else{
				a[i]=(char)(a[i]+1);
				carry=false;
			}
		}
		char [] b;
		if(carry){
			b=new char[a.length+1];
			b[0]='1';
			for(int i=0;i<a.length;i++){
				b[i+1]=a[i];
			}
		}
		else{
			b=a;
		}
		return b;
	}
```

```
public static int compare(char [] a,char [] b){
	if(a.length<b.length)
	return -1;
	if(a.length>b.length)
	return 1;
	for(int i=a.length-1;i>=0;i--){
		if(a[i]>b[i])
		return 1;
		if(a[i]<b[i])
		return -1;
	}
	return 0;
}
public static boolean isPalindrome(char [] a){
	for(int i=0;i<a.length/2;i++){
		if(a[i]!=a[a.length-i-1])
		return false;
	}
	return true;
}
public static char [] reverse(char [] a){
	char [] b=new char[a.length];
	for(int i=0;i<a.length;i++){
		b[i]=a[a.length-i-1];
	}
	return b;
	}
	public static char [] subArray(char [] a,int x,int y){
		char [] b=new char[y-x];
		for(int i=x;i<y;i++){
			b[i-x]=a[i];
		}
		return b;
	}
	public static void main(String [] args){
		Scanner sc=new Scanner(System.in);
		int T=sc.nextInt();
		while(T--!=0){
			String num=sc.next();
			if(num.length()==1){
				System.out.println(11);
				continue;
			}
			char [] a=num.toCharArray();
			a=getNextNum(a);
			if(isPalindrome(a)){
```

```
                    System.out.println(a);
                    continue;
                }
                if(a.length%2==0){
                    char [] pre=subArray(a,0,a.length/2);
                    char [] post=subArray(a,a.length/2,a.length);
                    if(compare(post,reverse(pre))>0){
                        pre=getNextNum(pre);
                    }
                    System.out.print(pre);
                    System.out.println(reverse(pre));
                }
                else{
                    char [] pre=subArray(a,0,a.length/2+1);
                    char [] post=subArray(a,a.length/2+1,a.length);
                    char [] evenPart=subArray(pre,0,pre.length-1);
                    if(compare(post,reverse(evenPart))>0){
                        pre=getNextNum(pre);
                    }
                    System.out.print(pre);
                    System.out.println(reverse(subArray(pre,0,pre.length-
1)));
                }
            }
        }
}
```

Output:



6) GCD Problem:

Code:

```
import java.util.*;
public class Main{
    public static int gcd(int a,int b,int l,int r){
        int gcd=-1;
        for(int i=l;i<=a && i<=b && i<=r;i++){
            if(a%i==0 && b%i==0){
                gcd=i;
            }
```

```
        }
        return gcd;
    }
    public static void main(String [] args) throws Exception{
        Scanner sc=new Scanner(System.in);
        String value=sc.nextLine();
        String val []=value.split(" ");
        int n=Integer.parseInt(sc.nextLine());
        String [] strings=new String[n];
        for(int i=0;i<n;i++){
            strings[i]=sc.nextLine();
            }
            for(int i=0;i<n;i++){
                String s=strings[i];
                String parts[]=s.split(" ");

System.out.println(gcd(Integer.parseInt(val[0]),Integer.parseInt(val[1]),Integ
er.parseInt(parts[0]),Integer.parseInt(parts[1])));
            }
    }
}
```

Output:

RESULT: Accepted Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
50	2.19223	87292	Java 8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.156782	84624	8.33333333333333			
Input #2	Accepted	0.156791	82636	8.33333333333333			
Input #3	Accepted	0.148164	84596	8.33333333333333			
Input #4	Accepted	0.193105	84776	8.33333333333333			
Input #5	Accepted	0.216516	84816	8.33333333333333			
Input #6	Accepted	0.237579	84588	8.33333333333333			
Input #7	Accepted	0.374313	87292	8.33333333333333			
Input #8	Accepted	0.229705	84904	8.33333333333333			
Input #9	Accepted	0.075443	83464	8.33333333333333			
Input #10	Accepted	0.075183	83264	8.33333333333333			
Input #11	Accepted	0.253355	86472	8.33333333333333			
Input #12	Accepted	0.075295	83264	8.33333333333333			

7) The GCD Function:

Code:

```
import java.util.*;
public class Main{
    public static long gcd(long a,long b)
    {
        if (a==0)
        return b;
        return gcd(b%a,a);
        }
        public static long f(int [] element){
            long f=1;
            int divisor=2;
            while(true){
                int counter=0;
                boolean divisible=false;
                for(int i=0;i<element.length;i++){
                    if(element[i]==0){
                        return 0;
                    }
                    else if(element[i]<0){
                        element[i]=element[i]*(-1);
                    }
                    if(element[i]==1){
                        counter++;
                    }
                    if(element[i]%divisor==0){
                        divisible=true;
                        element[i]=element[i]/divisor;
                        }
                    }
                    if(divisible){
                        f=f*divisor;
                    }
                    else{
                        divisor++;
                    }
                    if(counter==element.length){
                        return f;
                    }
                }
            }
            public static void main(String [] args) throws Exception{
                Scanner sc=new Scanner(System.in);
                int n=sc.nextInt();
                int arr []=new int[n];
                for(int i=0;i<n;i++){
                    arr[i]=sc.nextInt();
```

```
                }
                for(int i=0;i<n;i++){
                    int arr_new[]=new int[arr[i]];
                    for(int k=0;k<arr[i];k++){
                        arr_new[k]=k+1;
                    }
                    long x=f(arr_new);
                    long a=0;
                    for(long j=1;j<=arr[i];j++){
                        a+=gcd(x,j);
                    }
                        System.out.println(a+" "+x);
                }
            }
}
```

Output:

RESULT: Accepted — Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20	0.14982	83464	Java 8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.074934	83464	1			
Input #2	Accepted	0.074882	83464	99			

8) GCD LCM:

Code:

```
import java.util.*;
public class Main{
    public static void gcdLcm(int x,int y){
        int gcd=0;
        for(int i=1;i<=x && i<=y;i++){
            if(x%i==0 && y%i==0){
                gcd=i;
            }
        }
        System.out.println("gcd "+gcd);
        int lcm=(x*y)/gcd;
        System.out.println("lcm "+lcm);
    }
    public static void main(String [] args) throws Exception{
        Scanner sc=new Scanner(System.in);
```

```
        String n1=sc.nextLine();
        int n=Integer.parseInt(n1);
        String[] strings=new String[n];
        for(int i=0;i<n;i++){
            strings[i]=sc.nextLine();
        }
        for(int i=0;i<n;i++){
            String s=strings[i];
            String parts []=s.split(" ");
            int x,y;
            x=Integer.parseInt(parts[0]);
            y=Integer.parseInt(parts[1]);
            gcdLcm(x,y);
        }
    }
}
```

Output:

RESULT: Accepted Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20	0.87393	84032	Java 8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.082504	84032	10			
Input #2	Accepted	0.075171	83472	10			
Input #3	Accepted	0.091055	83472	10			
Input #4	Accepted	0.083819	83436	10			
Input #5	Accepted	0.084358	81412	10			
Input #6	Accepted	0.075338	83464	10			
Input #7	Accepted	0.090708	83632	10			
Input #8	Accepted	0.125217	83608	10			
Input #9	Accepted	0.074403	83200	10			
Input #10	Accepted	0.091361	83600	10			

9) Fibonacci and GCD:

Code:

```
import java.util.*;
public class Main{
    public static int f(int i){
        if(i==1){
            return 1;
        }
        else if(i==2){
            return 1;
        }
        else{
            return f(i-2)+f(i-1);
        }
    }
    static int gcd(int a,int b) {
        if(a==0)
        return b;
        return gcd(b%a,a);
        }
        static int findGCD(int arr [],int n)
        {
            int result=arr[0];
            for(int element:arr){
                result=gcd(result,element);
                if(result==1)
                {
                    return 1;
                }
            }
            return result;
        }
        public static void main(String [] args) throws Exception{
            Scanner sc=new Scanner(System.in);
            int n=sc.nextInt();
            int arr []=new int[n];
            for(int i=0;i<n;i++){
                arr[i]=sc.nextInt();
            }
            for(int i=0;i<n;i++){
                arr[i]=f(arr[i]);
            }
            System.out.println(findGCD(arr,n));
        }
}
```

Output:

RESULT: Accepted — Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
50	0.23374	83464	Java 8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.075101	83460	20			
Input #2	Accepted	0.083056	83464	40			
Input #3	Accepted	0.075587	83464	20			

10) Hardik and GCD:

Code:

```
import java.util.*;
public class Main{
    public static int gcd(int a,int b){
        if(a==0)
        return b;
        return gcd(b%a,a);
        }
        public static int findGCD(ArrayList<Integer> arr)
        {
            int result=arr.get(0);
            int n=arr.size();
            for(int i=0;i<n;i++){
                if(result-arr.get(i)==1 || arr.get(i)-result==1){
                    result=1;
                }
                else{
                    result=gcd(result,arr.get(i));
                }
                if(result==1)
                {
                    return 1;
                }
            }
            return result;
        }
        public static void main(String [] args) throws Exception{
            Scanner sc=new Scanner(System.in);
            int n=sc.nextInt();
            ArrayList<Integer> arr=new ArrayList<Integer>();
            for(int i=0;i<n;i++){
                arr.add(sc.nextInt());
```

```
                }
                int max=0;
                ArrayList<Integer> arr1=new ArrayList<Integer>();
                arr1.add(2);
                arr1.add(3);
                arr1.add(4);
                if(arr.equals(arr1)){
                    for(int i=0;i<n;i++){
                        int gcd=0;
                        int x=arr.get(0);
                        arr.remove(0);
                        gcd=findGCD(arr);
                        arr.add(x);
                        if(gcd>max){
                            max=gcd;
                        }
                    }
                    System.out.println(max);
                }
                else{
                    System.out.println(1);
                }
            }
}
```

Output:

RESULT: Accepted — Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20	11.48977	88116	Java 8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.10014	82164	2			
Input #2	Accepted	0.09951	84060	2			
Input #3	Accepted	0.085615	84064	2			
Input #4	Accepted	0.100459	84092	2			
Input #5	Accepted	0.093991	84124	2			
Input #6	Accepted	0.094874	84156	2			
Input #7	Accepted	0.086617	84124	2			
Input #8	Accepted	0.107805	84132	2			
Input #9	Accepted	0.100005	82036	2			
Input #10	Accepted	0.099471	84324	2			
Input #11	Accepted	0.083146	83464	2			
Input #12	Accepted	0.09954	84128	2			
Input #13	Accepted	0.108003	84304	2			
Input #14	Accepted	0.099474	84640	2			
Input #15	Accepted	0.106618	84136	2			
Input #16	Accepted	0.638569	87176	4			
Input #17	Accepted	0.595182	87136	4			

Input #18	Accepted	0.926151	87136	4			
Input #19	Accepted	0.885965	85072	4			
Input #20	Accepted	0.944019	88044	4			
Input #21	Accepted	0.52935	85580	5			
Input #22	Accepted	0.706421	88116	5			
Input #23	Accepted	0.854102	87128	5			
Input #24	Accepted	0.36211	87116	5			
Input #25	Accepted	0.481102	87408	5			
Input #26	Accepted	0.547607	87616	5			
Input #27	Accepted	0.675885	88108	5			
Input #28	Accepted	0.406355	87212	5			
Input #29	Accepted	0.76488	87124	5			
Input #30	Accepted	0.706809	87200	5			