Day 4:

SESSION 1:

Vertical Zig-Zag Pattern

ID:11072 Solved By 943 Users

The program must accept an integer **N** as the input. The program must print the desired pattern as shown in the Example Input/Output section.

Boundary Condition(s):
2 <= N <= 50

Input Format:
The first line contains N.

Output Format:
The first N lines contain the desired pattern as shown in the Example Input/Output section.

Example Input/Output 1:
Input:
5

Output:
1 10 11 20 21
2 9 12 19 22
3 8 13 18 23
4 7 14 17 24
5 6 15 16 25

Example Input/Output 2:
Input:
4

Output:
1 8 9 16
2 7 10 15
3 6 11 14
4 5 12 13

Max Execution Time Limit: 500 millisecs

Code:

```
import java.util.*;
public class zigzagpattern {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        for(int row=1;row<=N;row++){
            int val=row,dir=-1,down=(N-row)*2+1,up=(row-1)*2+1;
            for(int col=1;col<=N;col++){
                System.out.print(val+" ");
                if(dir==-1){
                    val+=down;
                }
```

```
                else{
                    val+=up;
                }
                dir*=-1;
            }
            System.out.println();
        }

    }
}
```

Stimulation:

https://pythontutor.com/visualize.html#code=%0Apublic%20class%20zigzagpattern%20%7B%0A%0A%20%20%20%20public%20static%20void%20main%28String%5B%5D%20args%29%20%7B%0A%20%20%20%20%20%20//Your%20Code%20Here%0A%20%20%20%20%0A%20%20%20%20%20%20%20%20int%20N%3D4%3B%0A%20%20%20%20%20%20%20%20for%28int%20row%3D1%3Brow%3C%3DN%3Brow%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20int%20val%3Drow,dir%3D-1,down%3D%28N-row%29*2%2B1,up%3D%28row-1%29*2%2B1%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20for%28int%20col%3D1%3Bcol%3C%3DN%3Bcol%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20System.out.print%28val%2B%22%20%22%29%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20if%28dir%3D%3D-1%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20val%2B%3Ddown%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20else%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20val%2B%3Dup%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20dir*%3D-1%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20System.out.println%28%29%3B%0A%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%7D%0A%7D&cumulative=false&curInstr=53&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=java&rawInputLstJSON=%5B%5D&textReferences=false

Problem 2:

Vertical Zig-Zag Reducing Pattern

ID:11074 Solved By 930 Users

The program must accept an integer **N** as the input. The program must print the desired pattern as shown in the Example Input/Output section.

Boundary Condition(s):
2 <= N <= 50

Input Format:
The first line contains N.

Output Format:
The first N lines contain the desired pattern as shown in the Example Input/Output section.

Example Input/Output 1:
Input:
5

Output:
1
2 9
3 8 10
4 7 11 14
5 6 12 13 15

Example Input/Output 2:
Input:
6

Output:
1
2 11
3 10 12
4 9 13 18
5 8 14 17 19
6 7 15 16 20 21

Max Execution Time Limit: 500 millisecs

Code:

```
import java.util.*;
public class zigzagReducing {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in= new Scanner(System.in);
        int N=in.nextInt();
        for(int row=1;row<=N;row++){
            int val=row,dir=-1,down=(N-row)*2+1;
            for(int col=1;col<=row;col++){
                System.out.print(val+" ");
                if(dir==-1){
```

```
                    val+=down;
                }
                else{
                    val+=(row-col)*2;
                }
                dir*=-1;
            }
            System.out.println();
        }

    }
}
```

Stimulation:

https://pythontutor.com/visualize.html#code=%0Apublic%20class%20zigzagReducing%20%7B%0A%0A%20%20%20%20public%20static%20void%20main%28String%5B%5D%20args%29%20%7B%0A%0A%20%20%20%20%20%20%20%20int%20N%3D5%3B%0A%20%20%20%20%20%20%20%20for%28int%20row%3D1%3Brow%3C%3DN%3Brow%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20int%20val%3Drow,dir%3D-1,down%3D%28N-row%29*2%2B1%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20for%28int%20col%3D1%3Bcol%3C%3Drow%3Bcol%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20System.out.print%28val%2B%22%20%22%29%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20if%28dir%3D%3D-1%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20val%2B%3Ddown%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20else%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20val%2B%3D%28row-col%29*2%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20dir*%3D-1%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20System.out.println%28%29%3B%0A%20%20%20%20%20%20%20%20%7D%0A%0A%20%20%20%20%7D%0A%7D&cumulative=false&curInstr=37&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=java&rawInputLstJSON=%5B%5D&textReferences=false

SESSION 2:

String Characters - Combinations Sorted

ID:11075 Solved By 798 Users

The program must accept a string **S** as the input. The program must print the combinations of the characters in the string S in sorted order.

Boundary Condition(s):
2 <= Length of S <= 15

Input Format:
The first line contains the string S.

Output Format:
The lines containing the string values representing the combinations of the characters in the string S in sorted order.

Example Input/Output 1:
Input:
abc

Output:
a
ab
abc
ac
b
bc
c

Example Input/Output 2:
Input:
virus

Output:
i
ir
irs
iru
irus
is
iu
ius
r
rs
ru
rus
s
u
us
v
vi
vir
virs
viru
virus
vis
viu
vius
vr
vrs
vru
vrus
vs
vu
vus

Max Execution Time Limit: 1000 millisecs

Code:

```
import java.util.*;
public class combinationSorted {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        String str= in.nextLine();
        List<String> values = new ArrayList<>();
        for(int ctr=1;ctr<(1<<str.length());ctr++){
            StringBuilder sb= new StringBuilder();
            for(int bmi=0;bmi<str.length();bmi++){
```

```
                if((ctr&(1<<bmi)) !=0){
                    sb.append(str.charAt(bmi));
                }
            }
            values.add(sb.toString());
        }
        Collections.sort(values);
        for(String val:values){
            System.out.println(val);
        }
    }
}
```

Problem 2:

Combination Zero Sum

ID:11076 Solved By 817 Users

The program must accept **N** integers as the input. The program must print the number of combinations of the integers (among the N integers) which add up to 0 as the output.

Boundary Condition(s):
2 <= N <= 15
-100 <= Each integer value <= 100

Input Format:
The first line contains N.
The second line contains N integers separated by a space.

Output Format:
The first line contains the number of combinations of the integers (among the N integers) which add up to 0.

Example Input/Output 1:
Input:
5
10 -5 5 -15 20

Output:
3

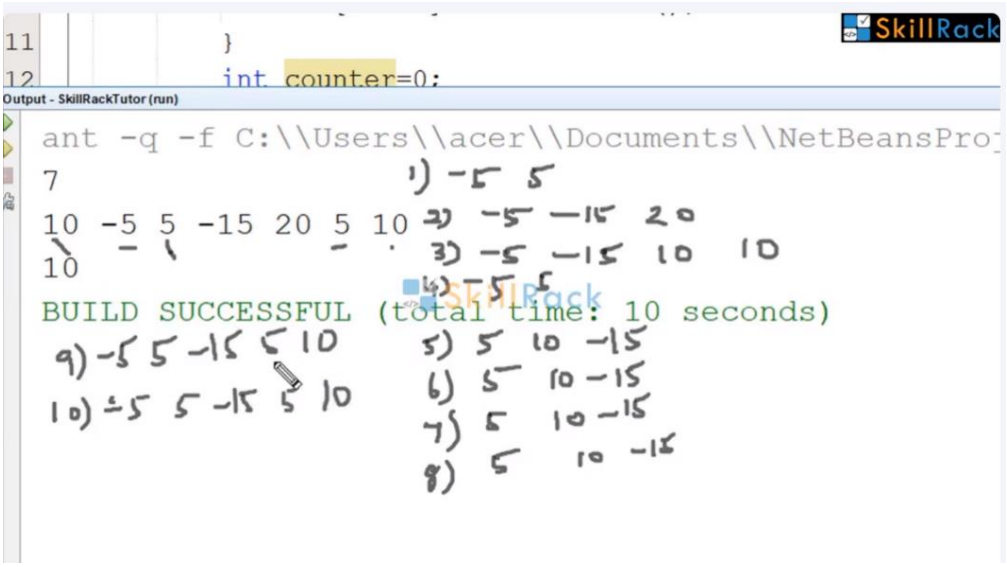
Explanation:
The three combinations which add up to 0 are
10, 5, -15
20, -5, -15
-5, 5

Example Input/Output 2:
Input:
7
10 -5 5 -15 20 5 10

Output:
10

Max Execution Time Limit: 500 millisecs

Logic:



Code:

```java
import java.util.*;
public class combinationZeroSum {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        int arr[] = new int[N];
        for(int i=0;i<N;i++){
            arr[i]=in.nextInt();
        }
        int counter=0;
        for(int ctr=1;ctr<(1<<N);ctr++){
            int sum=0;
            for(int bmi=0;bmi<N;bmi++){
                if((ctr&(1<<bmi))!=0){
                    sum+=arr[bmi];
                }
            }
            if(sum==0){
                counter++;
            }
        }
        System.out.println(counter);
    }
```

```
}
```

Stimulation:

https://pythontutor.com/visualize.html#code=%0Apublic%20class%20combinationZeroSum%20%7B%0A%0A%20%20%20%20public%20static%20void%20main%28String%5B%5D%20args%29%20%7B%0A%0A%20%20%20%20%20%20%20%20int%20N%3D5%3B%0A%20%20%20%20%20%20%20%20int%20arr%5B%5D%20%3D%7B10,-5,5,-15,20%7D%3B%0A%0A%20%20%20%20%20%20%20%20int%20counter%3D0%3B%0A%20%20%20%20%20%20%20%20for%28int%20ctr%3D1%3Bctr%3C%281%3C%3CN%29%3Bctr%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20int%20sum%3D0%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20for%28int%20bmi%3D0%3Bbmi%3CN%3Bbmi%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20if%28%28ctr%26%281%3C%3Cbmi%29%29!%3D0%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20sum%2B%3Darr%5Bbmi%5D%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%20%20%20%20if%28sum%3D%3D0%29%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20counter%2B%2B%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%20%20%20%20System.out.println%28counter%29%3B%0A%20%20%20%20%7D%0A%7D&cumulative=false&curInstr=60&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=java&rawInputLstJSON=%5B%5D&textReferences=false

SESSION 3:

Problem 1:

Find Single Value Repeated Odd Times

ID:11077 Solved By 949 Users

The program must accept **N** integers (where N is always odd) as the input. Among the N integers, all the integers have occurred even number of times except one integer X. The program must find and print the integer X as the output.

Boundary Condition(s):
3 <= N <= 99999
1 <= Each integer value <= 10^8

Input Format:
The first line contains the integer N.
The second line contains N integers separated by a space.

Output Format:
The first line contains the integer X.

Example Input/Output 1:
Input:
5
44 54 88 44 54

Output:
88

Explanation:
The integers 44 and 54 have occurred twice (even number of times).
The integer 88 has occurred only once (odd number of times).
Hence 88 is printed as the output.

Example Input/Output 2:
Input:
7
55 55 55 55 55 55 55

Output:
55

Max Execution Time Limit: 200 millisecs

Code:

```
import java.util.*;
public class singleValueRepeatedOddTimes {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();

        int arr[] = new int[N];

        for(int i=0;i<N;i++){
```

```
            arr[i]=in.nextInt();
        }

        int oddnum=0;
        for(int i=0;i<N;i++){
            oddnum=oddnum^arr[i];
        }
        System.out.println(oddnum);
    }
}
```

Problem 2:

Count 1s in Binary Representation of N

ID:9491 Solved By 1030 Users

The program must accept an integer **N** as the input. The program must print the number of 1s in the binary representation of N as the output.

Boundary Condition(s):
1 <= N <= 10^8

Input Format:
The first line contains N.

Output Format:
The first line contains the count of 1s in the binary representation of N.

Example Input/Output 1:
Input:
10

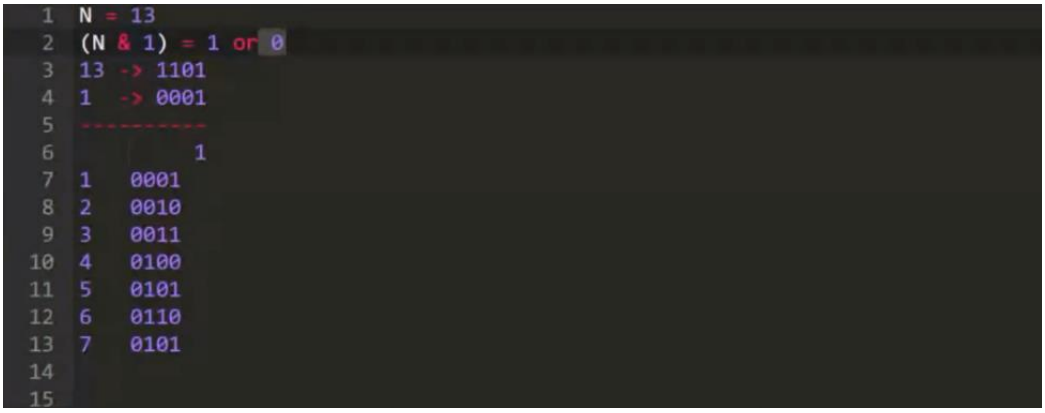
Output:
2

Explanation:
The binary representation of **10** is **1010**. So there are two 1s in 1010.
Hence the output is 2

Example Input/Output 2:
Input:
15

Output:
4

Max Execution Time Limit: 1000 millisecs



Code:

```
import java.util.*;
public class BinaryOneCount {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in=new Scanner(System.in);
        int N=in.nextInt();
        int counter=0;
        while(N!=0){
            if((N&1)==1){
                counter++;
            }
            N=N>>1; // n>>1 equivalent to n=n/2

        }
        System.out.println(counter);
    }
}
```

Problem 3:

Flip Bits Count

ID:5160 Solved By 1402 Users

The program must accept two integers **A** and **B** as the input. The program must print the number of bits to be flipped to convert A to B as the output.

Boundary Condition(s):
2 <= A, B <= 10^8

Input Format:
The first line contains A and B separated by a space.

Output Format:
The first line contains the number of bits to be flipped to convert A to B.

Example Input/Output 1:
Input:
12 10

Output:
2

Explanation:
The binary representation of 12 is 1100.
The binary representation of 10 is 1010.
After flipping the middle 2 bits in the binary representation of 12, it becomes the binary representation of 10.
So 2 is printed as the output.

Example Input/Output 2:
Input:
10 20

Output:
4

Max Execution Time Limit: 500 millisecs

Code:

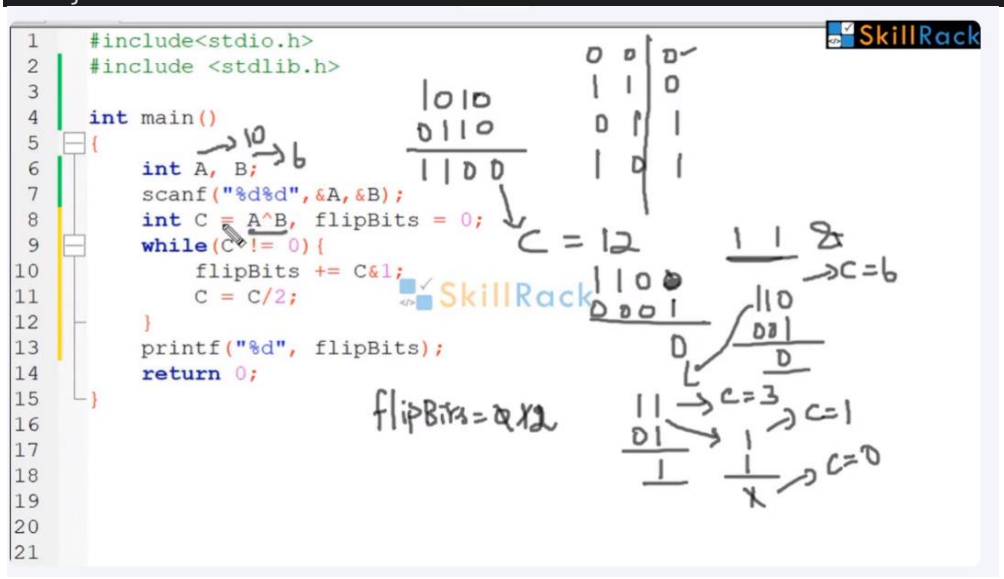
```java
import java.util.*;

public class flipBitsBinary {
        public static void main(String[] args) {
            //Your Code Here
            Scanner in =new Scanner(System.in);
            int A=in.nextInt(),B=in.nextInt();
            int C = A^B, flipBits=0;
            while(C!=0){
                flipBits+=C&1;
                C=C/2;
            }
```

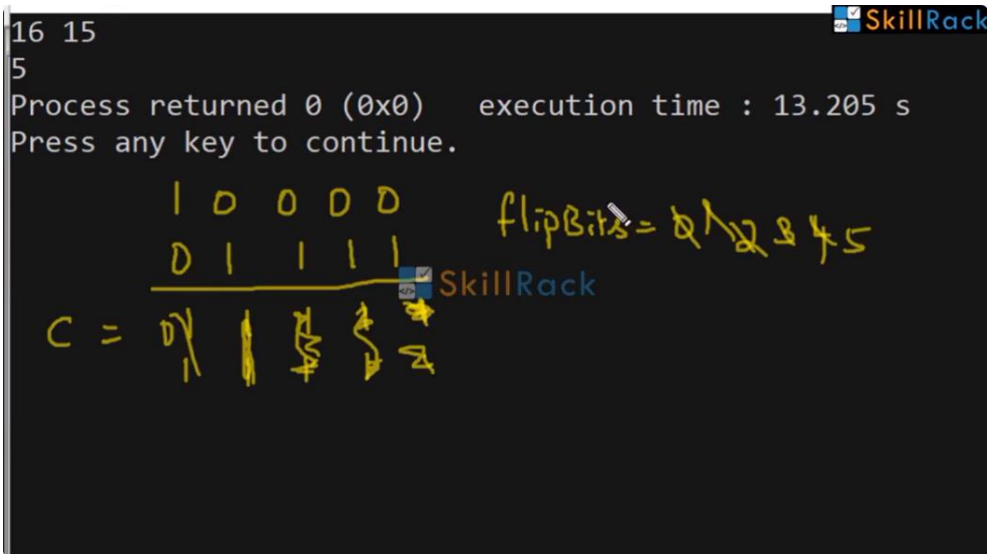
```
            System.out.println(flipBits);
        }
    }
```

```c
#include<stdio.h>
#include <stdlib.h>

int main()
{
    int A, B;
    scanf("%d%d",&A,&B);
    int C = A^B, flipBits = 0;
    while(C != 0){
        flipBits += C&1;
        C = C/2;
    }
    printf("%d", flipBits);
    return 0;
}
```

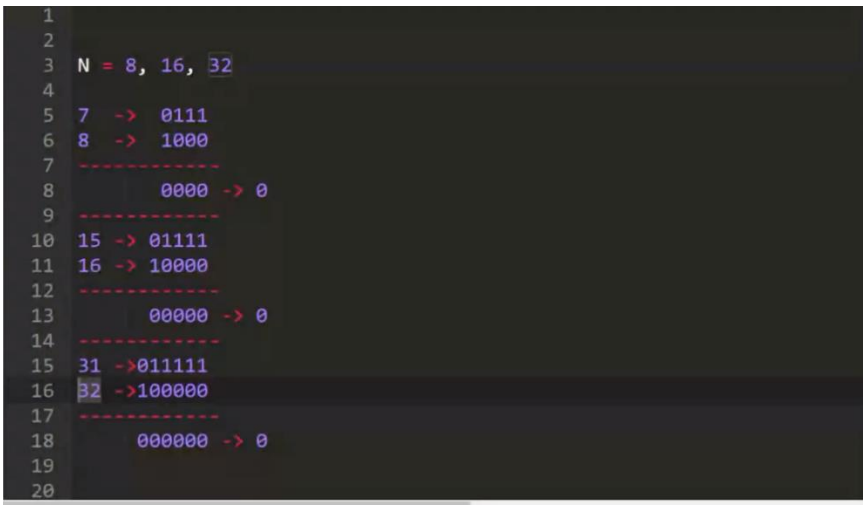



```
16 15
5
Process returned 0 (0x0)   execution time : 13.205 s
Press any key to continue.
```



PROBLEM: FIND WHETHER THE GIVEN NUMBER IS POWER OF TWO OR NOT:

Logic:



Code:

```
import java.util.Scanner;

public class findWhetherNumberPowerOfTWO {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        System.out.println((N&(N-1)) ==0 ? "Yes, the given number is a power of
two(2)":"No, The given number is not the power of two(2)");
    }
}
```

If the value of N and the previous value of N is equal to zero then the given number is a power of two