CATH CON 2020 Questions

Senior Category Answers:

Up

```
import java.util.*;
class Up
{
	public static void main(String[] args)
	{
		Scanner sc = new Scanner(System.in);
		int r = sc.nextInt();
		int a = (int)(0.3*r);
		for(int i = 0; i<r/2+1; i++)
		{
			for(int j = i; j<r/2; j++)
			{
				System.out.print(" ");
			}
			for(int k = i; k>=0; k--)
			{
			  System.out.print("*");
			}
			for(int x = i; x>=1; x--)
			{
				System.out.print("*");
			}
			System.out.println();
		}

		for(int i = 0; i<r; i++)
		{
			for(int j = 0; j<r-a; j++)
			{
				if(j<a)
				{
					System.out.print(" ");
				}
				else
				{
					System.out.print("*");
				}
			}
```

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

Roman Numeral

```
import java.util.*;
class RomanNumeral
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a Roman Number");
        String s = sc.nextLine();
        String check = "MDCLXVI";
        int c = 0;
        if(s.length() == 1)
        {
            c = ReturnDigit(s.charAt(0));
        }
        else
        {
            int i = 0;
            for(; i<s.length()-1; i++)
            {
                String x = s.substring(i,i+2);
                if(check.indexOf(x.charAt(0)) > check.indexOf(x.charAt(1)))
                {
                    c+= (ReturnDigit(x.charAt(1)) - ReturnDigit(x.charAt(0)));
                    i++;
                }
                else
                {
                    c+=ReturnDigit(x.charAt(0));
                }
            }
            if(i==s.length()-1)
            {
                c+=ReturnDigit(s.charAt(i));
            }
        }
        System.out.println(c);
    }
```

```
	public static int ReturnDigit(char ch)
	{
		switch(ch){
			case 'M':
				return 1000;
			case 'D':
				return 500;
			case 'C':
				return 100;
			case 'L':
				return 50;
			case 'X':
				return 10;
			case 'V':
				return 5;
			case 'I':
				return 1;
			default:
				return 0;
		}
	}
}
```

Chess

```
import java.util.*;
class Chess
{
	static int si,sj,fi,fj;
	static char board[][] = new char[8][8];
	public static void main(String[] args)
	{
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter Position of starting point");
		si = sc.nextInt();
		sj = sc.nextInt();
		System.out.println("Enter Position of ending point");
		fi = sc.nextInt();
		fj = sc.nextInt();
		boolean g = true;

		if(inRange(si) && inRange(sj) && inRange(fi) && inRange(fj))
		{
```

```
for(int i = 0; i<8; i++)
{
	for(int j = 0; j<8; j++)
	{
		board[i][j] = '.';
	}
}

board[si][sj] = 'S';
board[fi][fj] = 'F';

for(int i = 0; i<8; i++)
{
	for(int j = 0; j<8; j++)
	{
		System.out.print(board[i][j] + " ");
	}
	System.out.println();
}
if(isKing())
{
	System.out.println("King");
	g = false;
}
if(isQueen())
{
	System.out.println("Queen");
	g = false;
}
if(isBishop())
{
	System.out.println("Bishop");
	g = false;
}
if(isKnight())
{
	System.out.println("Knight");
	g = false;
}
if(isRook())
{
	System.out.println("Rook");
	g = false;
}
```

```
                if(isPawn())
                {
                        System.out.println("Pawn");
                        g = false;
                }
                if(g)
                {
                        System.out.println("No Piece");
                }
        }
        else
        {
                System.out.println("Index does not exist on board");
        }
}

static boolean isKing()
{
        int a = si-fi;
        int b = sj-fj;
        if((-1<=a && a<=1) && (-1<=b && b<=1))
        {
                return true;
        }

        return false;
}

static boolean isQueen()
{
        if(isRook() || isBishop())
        {
                return true;
        }

        return false;
}

static boolean isBishop()
{
        if(((si+sj) == (fi+fj)) || (si-fi) == (sj-fj))
        {
                return true;
        }
```

```
        return false;
}

static boolean isKnight()
{
        int a = si-fi;
        int b = sj-fj;
        if((a==1 || a==-1) && (b==2 || b==-2))
        {
                return true;
        }
        if((a==2 || a==-2) && (b==1 || b==-1))
        {
                return true;
        }

        return false;
}

static boolean isRook()
{
        if((si == fi) || (sj == fj))
        {
                return true;
        }

        return false;
}

static boolean isPawn()
{
        if(sj == fj && si-fi == 1)
        {
                return true;
        }

        return false;
}

static boolean inRange(int n)
{
        if(0<=n && n<8)
        {
```

```
                return true;
            }

            return false;
        }
}
```

Concentric Characters

```
import java.util.*;
class ConcentricCharacters
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter n");
        int n = sc.nextInt();
        int arr[][] = new int[2*n-1][2*n-1];

        for(int i = 0; i<(2*n-1); i++)
        {
            for(int j = 0; j<(2*n-1); j++)
            {
                arr[i][j] = 0;
            }
        }

        for(int x = n; x>=1; x--)
        {
            for(int i = 0; i<(2*n-1); i++)
            {
                for(int j = 0; j<(2*n-1); j++)
                {
                    if(((i == n-x || j == n-x) || (i == n+x-2 || j == n+x-2)) &&
arr[i][j] == 0)
                    {
                        arr[i][j] = x;
                    }
                }
            }
        }

        for(int i = 0; i<(2*n-1); i++)
        {
```

```
                for(int j = 0; j<(2*n-1); j++)
                {
                        System.out.print(arr[i][j]);
                }
                System.out.println();
            }
        }
}
```

Fibonacci Alphabets

```
import java.util.*;
class LetterFib
{
        public static void main(String[] args)
        {
                int a = 1;
                int b = 1;
                Scanner sc = new Scanner(System.in);
                int n = sc.nextInt();
                char arr[] = new char[findArrLen(n)];
                int i = 0;
                for(i = 0; i<arr.length;)
                {
                        arr[i] = (char)(65+((a-1)%26));
                        i++;
                        a+=b;
                        if(i<arr.length)
                        {
                                arr[i] = (char)(65+((b-1)%26));
                                i++;
                                b+=a;
                        }
                        else
                        {
                                break;
                        }
                }

                int k = 0;

                for(i = 1; i<=n; i++)
                {
                        for(int j = 0; j<i; j++)
```

```
                        {
                                System.out.print(arr[k]);
                                k++;
                        }
                        System.out.println();
                }
        }

        static int findArrLen(int x)
        {
                int sum = 0;
                for(int i = 1; i<=x; i++)
                {
                        sum+=i;
                }

                return sum;
        }
}
```