

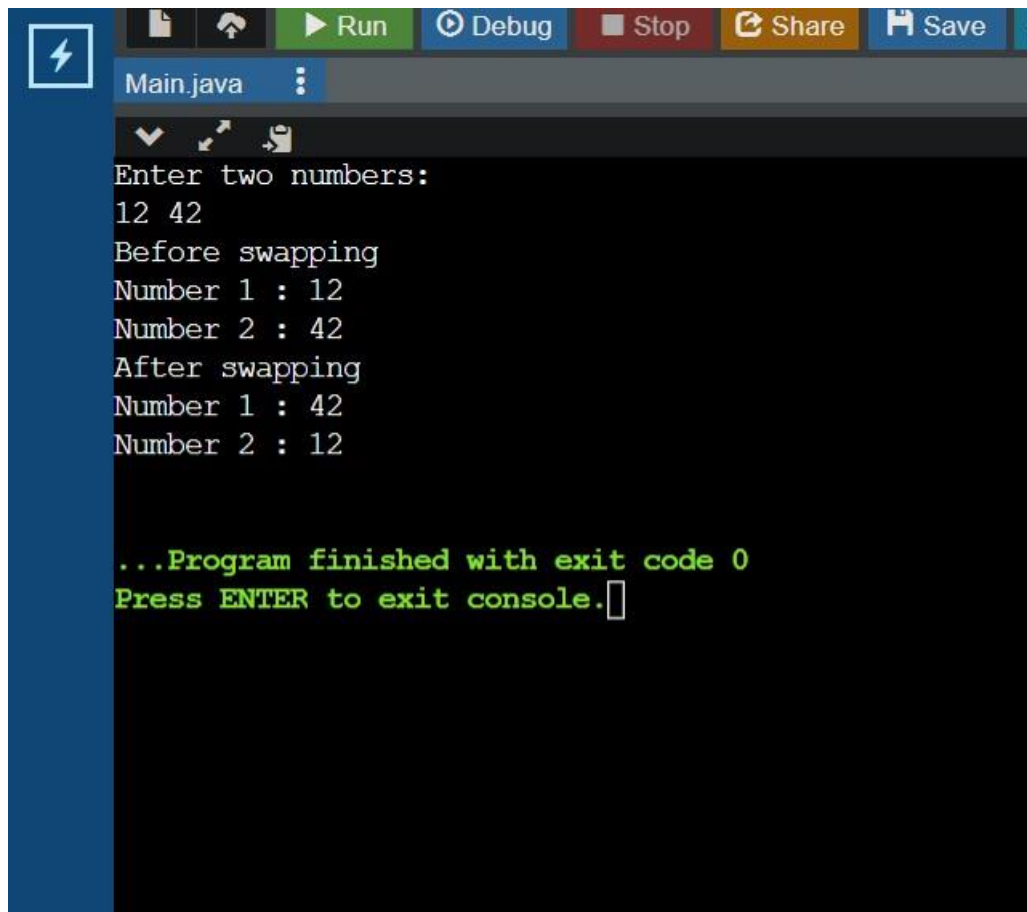
Q1. Write a program in java to swap two numbers.

CODE:

```
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter two numbers: ");
        int n1=sc.nextInt();
        int n2=sc.nextInt();
        System.out.println("Before swapping");
        System.out.println("Number 1 : "+n1+"\nNumber 2 : "+n2);
        int temp=n1;
        n1=n2;
        n2=temp;
        System.out.println("After swapping");
        System.out.println("Number 1 : "+n1+"\nNumber 2 : "+n2);
    }
}
```

OUTPUT:



The screenshot shows an IDE interface with a toolbar at the top containing icons for file operations and buttons for Run, Debug, Stop, Share, and Save. Below the toolbar, a tab labeled 'Main.java' is open. The main area of the IDE is a dark-themed console window displaying the output of a Java program. The program prompts the user to 'Enter two numbers:', and the input '12 42' is shown. It then prints 'Before swapping', followed by 'Number 1 : 12' and 'Number 2 : 42'. After performing a swap, it prints 'After swapping', followed by 'Number 1 : 42' and 'Number 2 : 12'. At the bottom, a green message states '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor.

```
Enter two numbers:
12 42
Before swapping
Number 1 : 12
Number 2 : 42
After swapping
Number 1 : 42
Number 2 : 12

...Program finished with exit code 0
Press ENTER to exit console.
```

Q2. Write a java program to check whether given number is even or odd.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int a=sc.nextInt();

        if(a%2==0)

            System.out.print("Entered number is even");

        else

            System.out.print("Entered number is odd");

    }

}
```



```
Enter a number: 87
Entered number is odd

...Program finished with exit code 0
Press ENTER to exit console.
```

OUTPUT:

Q3. Write a java program to find factorial of a number.

CODE:

```
import java.util.*;

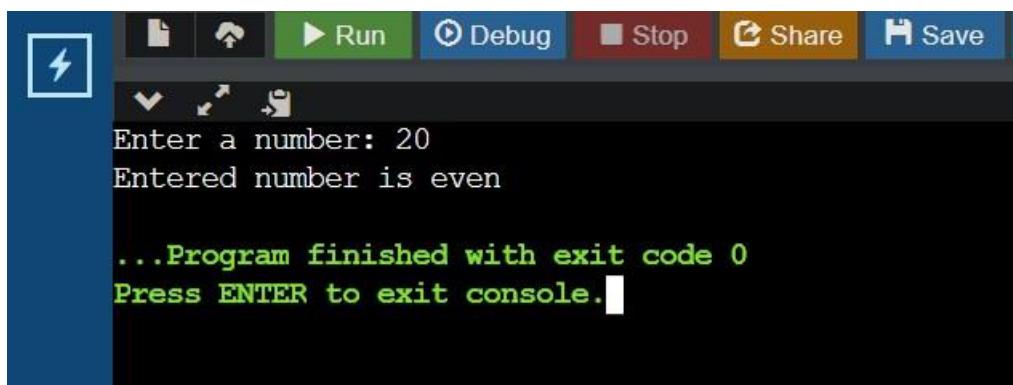
public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int n=sc.nextInt();
```

A screenshot of a Java IDE's console window. The window has a dark background with a toolbar at the top containing icons for Run (green), Debug (blue), Stop (red), Share (orange), and Save (blue). Below the toolbar, the console output shows the program's execution: "Enter a number: 20", "Entered number is even", and "...Program finished with exit code 0". The prompt "Press ENTER to exit console." is visible at the bottom of the console area, followed by a white cursor line.

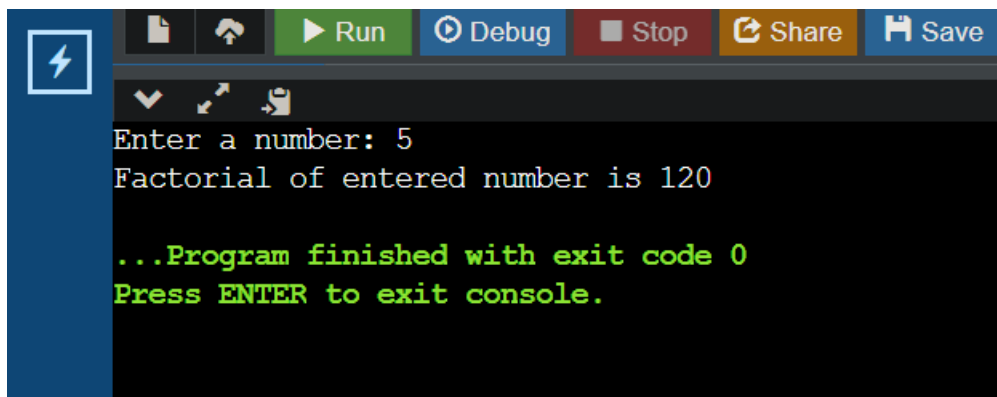
```
Enter a number: 20
Entered number is even

...Program finished with exit code 0
Press ENTER to exit console.
```

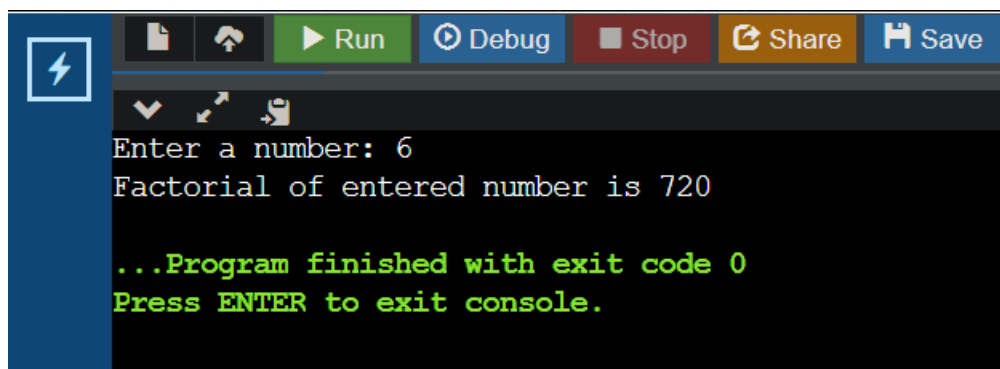
```
int fact=1;
```

```
for(int i=1;i<=n;i++)  
    fact=fact*i;  
System.out.print("Factorial of entered number is "+fact);  
}  
}
```

OUTPUT:



The screenshot shows a Java IDE interface with a toolbar at the top containing icons for file operations and buttons for Run, Debug, Stop, Share, and Save. The console area displays the following text: "Enter a number: 5", "Factorial of entered number is 120", "...Program finished with exit code 0", and "Press ENTER to exit console." in green text.



The screenshot shows the same Java IDE interface as above. The console area displays the following text: "Enter a number: 6", "Factorial of entered number is 720", "...Program finished with exit code 0", and "Press ENTER to exit console." in green text.

Q4. Using a switch statement, write a menu driven program in java to calculate maturity amount of a bank deposit.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Type 1 for Term Deposit");
        System.out.println("Type 2 for Recurring Deposit");
        System.out.print("Enter your choice: ");

        int ch = sc.nextInt();

        double p = 0.0, r = 0.0, a = 0.0;

        int n = 0;

        switch (ch) {

            case 1: System.out.print("Enter Principal: ");

                p = sc.nextDouble();

                System.out.print("Enter Interest Rate: ");

                r = sc.nextDouble();

                System.out.print("Enter time in years: ");

                n = sc.nextInt();

                a = p * Math.pow(1 + r / 100, n);

                System.out.println("Maturity amount = " + a);

                break;

            case 2: System.out.print("Enter Monthly Installment: ");

                p = sc.nextDouble();

                System.out.print("Enter Interest Rate: ");

                r = sc.nextDouble();

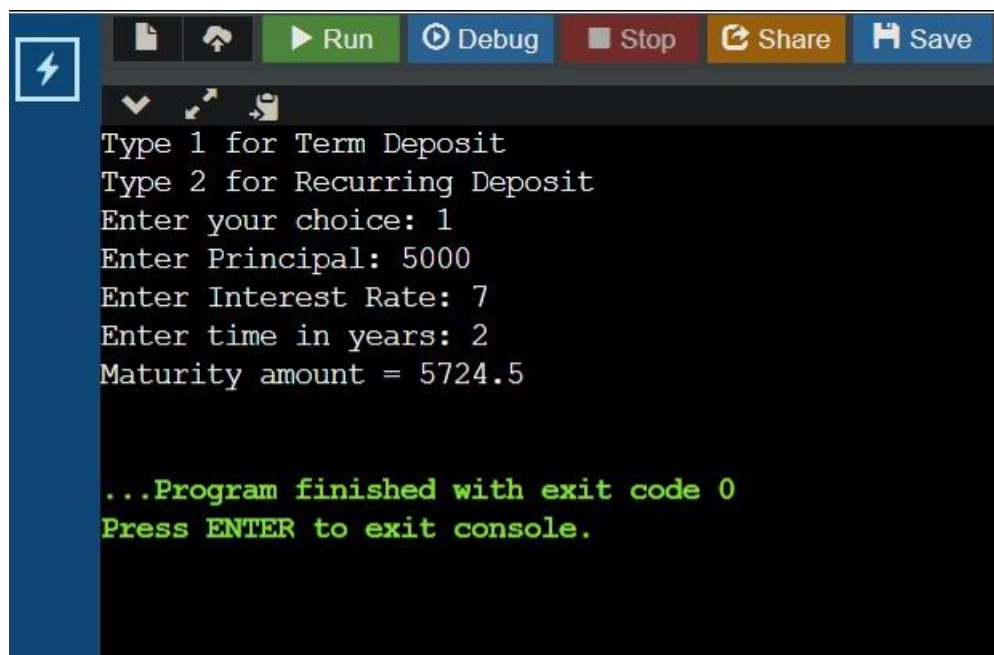
                System.out.print("Enter time in months: ");
```

```

        n = sc.nextInt();
        a = p*n + p*((n*(n+1))/2) * (r/100) * (1/12.0);
        System.out.println("Maturity amount = " + a);
        break;
    default: System.out.println("Invalid choice");
}
}
}

```

OUTPUT:



The screenshot shows a Java IDE window with a toolbar at the top containing icons for Run, Debug, Stop, Share, and Save. The console output is as follows:

```

Type 1 for Term Deposit
Type 2 for Recurring Deposit
Enter your choice: 1
Enter Principal: 5000
Enter Interest Rate: 7
Enter time in years: 2
Maturity amount = 5724.5

...Program finished with exit code 0
Press ENTER to exit console.

```

Q5. Write a java program to convert kilometres to miles.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String []args) {

        Scanner sc =new Scanner(System.in);

        System.out.print("Enter the distance in kilometer : ");

        int n = sc.nextInt();

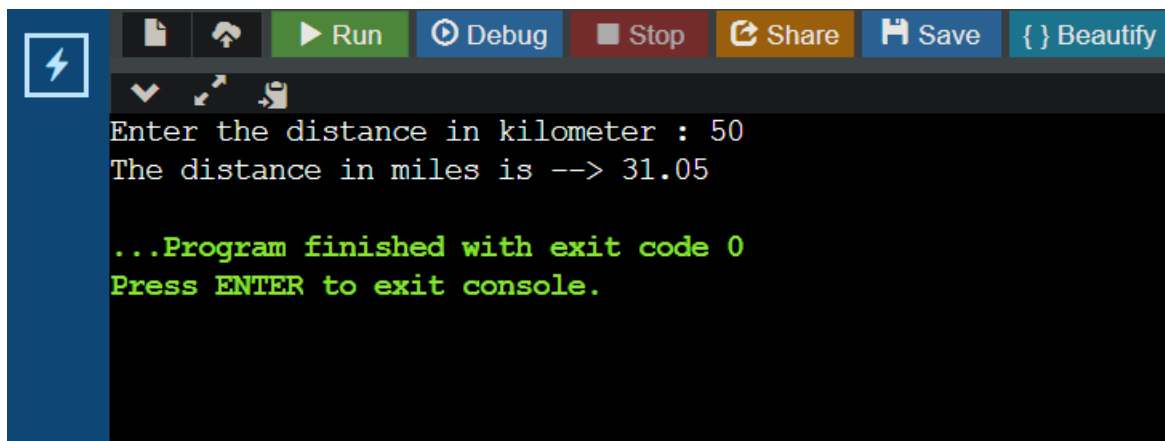
        double miles = n * 0.621;

        System.out.print("The distance in miles is --> " + miles);

    }

}
```

OUTPUT:

A screenshot of a Java IDE's console window. The window has a dark background with a toolbar at the top containing icons for Run (a green play button), Debug (a blue play button with a magnifying glass), Stop (a red square), Share (a blue square with a share icon), Save (a blue floppy disk icon), and Beautify (a blue square with curly braces). Below the toolbar, the console output is displayed in a monospaced font. It shows the program prompting for input, receiving '50', and outputting '31.05'. The program then finishes with exit code 0, and a message prompts the user to press ENTER to exit the console.

```
Enter the distance in kilometer : 50
The distance in miles is --> 31.05

...Program finished with exit code 0
Press ENTER to exit console.
```


Q6. Write a java program to find whether the given number are friendly pair or not.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter two numbers: ");

        int n1=sc.nextInt();

        int n2=sc.nextInt();

        int s1=0,s2=0;

        for(int i=1;i<n1;i++) {

            if(n1%i==0)

                s1=s1+i;

        }

        for(int i=1;i<n2;i++) {

            if(n2%i==0)

                s2=s2+i;

        }

        if(s1==n1&& s2==n2)

            System.out.print("Entered numbers are friendly pair");

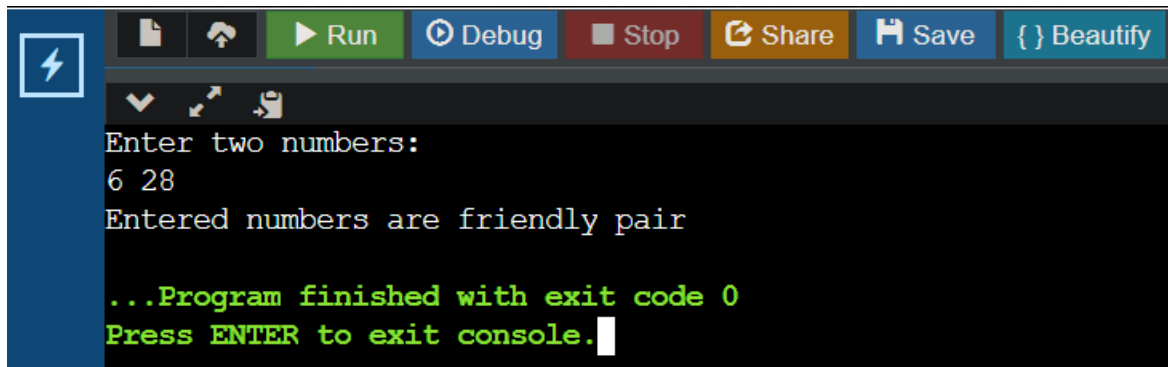
        else

            System.out.print("Entered numbers are not friendly pairs");

    }

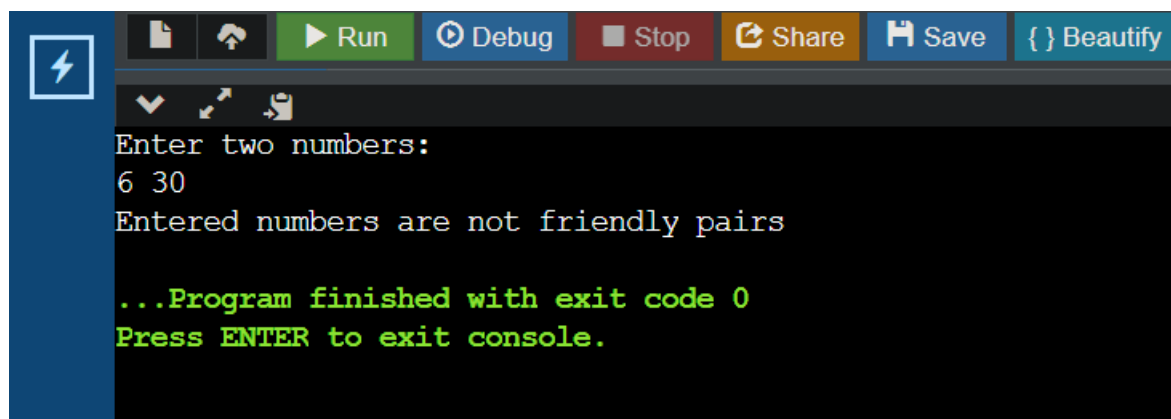
}
```

OUTPUT:



```
Enter two numbers:
6 28
Entered numbers are friendly pair

...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter two numbers:
6 30
Entered numbers are not friendly pairs

...Program finished with exit code 0
Press ENTER to exit console.
```

Q7. Write a java program to replace all 0's with 1's in an integer number.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number : ");

        int a = sc.nextInt();

        String str = Integer.toString(a);

        String str1 = "";

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

            if (str.charAt(i) == '0')

                str1 = str1 + '1';

            else

                str1 = str1 + str.charAt(i);

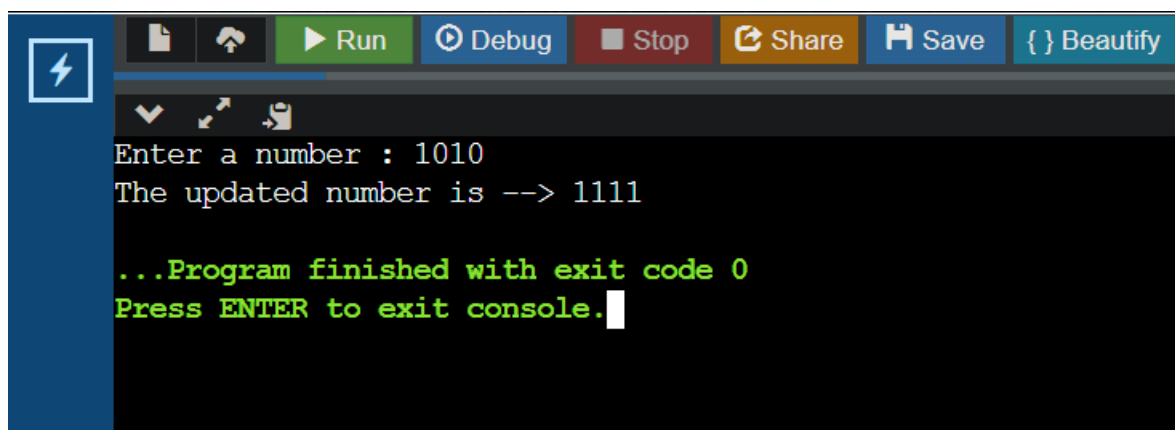
        }

        System.out.print("The updated number is --> " + str1);

    }

}
```

OUTPUT:



```
Enter a number : 1010
The updated number is --> 1111

...Program finished with exit code 0
Press ENTER to exit console.
```

Q8. Write a java program to print an array into zig zag fashion.

CODE:

```
import java.util.Scanner;

public class Main {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int[] array;

        int length = 0;

        System.out.print("Enter the number of elements in array: ");

        length = sc.nextInt();

        array = new int[length];

        System.out.print("Enter the " + length + " elements: ");

        for (int i = 0; i < length; i++)

            array[i] = sc.nextInt();

        System.out.print("\nThe zigzag Array is: \n");

        zigzagArray(array, length);

        for (int i = 0; i < length; i++)

            System.out.print(+array[i] + " ");

        System.out.print("\n");

    }

    public static void zigzagArray(int array[], int n) {

        boolean flag = true;

        int temp;

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

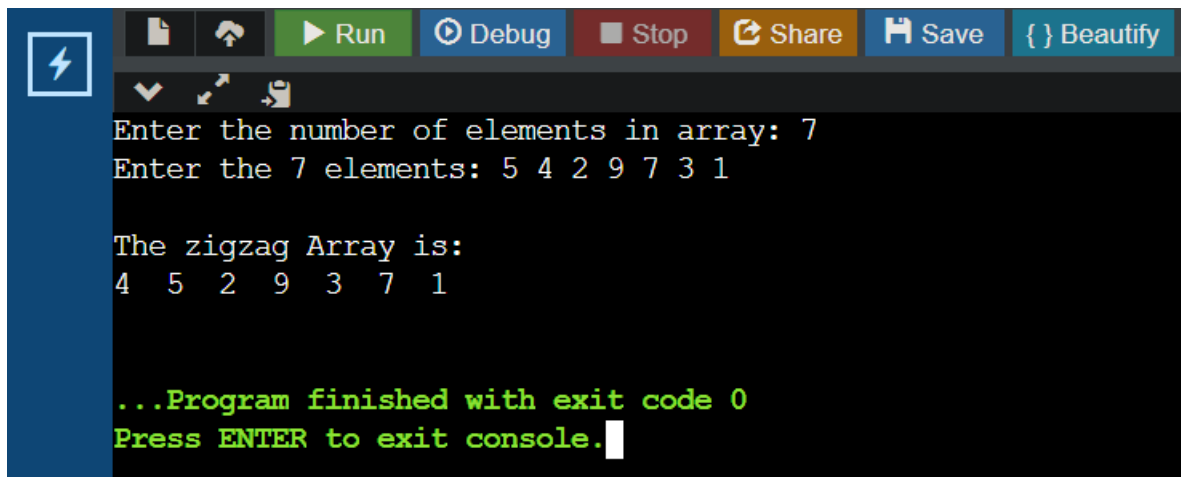
            if (flag) {

                if (array[i] > array[i + 1]) {
```

```

        temp = array[i];
        array[i] = array[i + 1];
        array[i + 1] = temp;
    }
}
else {
    if (array[i] < array[i + 1]) {
        temp = array[i];
        array[i] = array[i + 1];
        array[i + 1] = temp;
    }
}
flag = !flag;
}
}
}

```



```

Enter the number of elements in array: 7
Enter the 7 elements: 5 4 2 9 7 3 1

The zigzag Array is:
4 5 2 9 3 7 1

...Program finished with exit code 0
Press ENTER to exit console.

```

OUTPUT:

Q9. Write a java program to sort n elements in 1-D array using merge sort.

CODE:

```
public class Main {  
    void merge(int arr[], int l, int m, int r) {  
        int n1 = m - l + 1;  
        int n2 = r - m;  
        int L[] = new int [n1];  
        int R[] = new int [n2];  
        for (int i=0; i<n1; ++i)  
            L[i] = arr[l + i];  
        for (int j=0; j<n2; ++j)  
            R[j] = arr[m + 1+ j];  
        int i = 0, j = 0;  
        int k = l;  
        while (i < n1 && j < n2) {  
            if (L[i] <= R[j]) {  
                arr[k] = L[i];  
                i++;  
            }  
            else {  
                arr[k] = R[j];  
                j++;  
            }  
            k++;  
        }  
        while (i < n1) {  
            arr[k] = L[i];
```

```

        i++;
        k++;
    }
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void sort(int arr[], int l, int r) {
    if (l < r) {
        int m = (l+r)/2;
        sort(arr, l, m);
        sort(arr, m+1, r);
        merge(arr, l, m, r);
    }
}

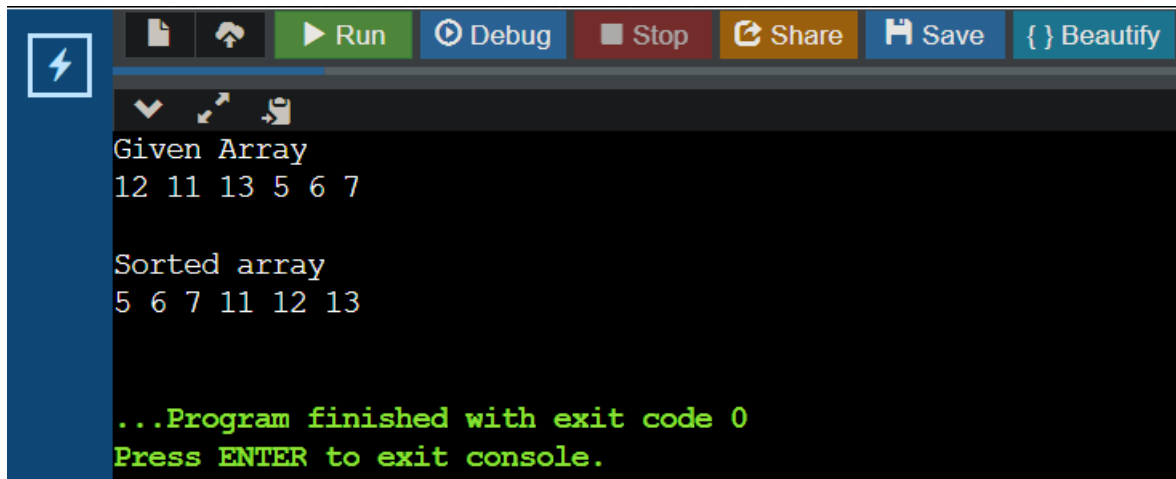
static void printArray(int arr[]) {
    int n = arr.length;
    for (int i=0; i<n; ++i)
        System.out.print(arr[i] + " ");
    System.out.println();
}

public static void main(String args[]) {
    int arr[] = { 12, 11, 13, 5, 6, 7 };
    System.out.println("Given Array");
    printArray(arr);
}

```

```
Main ob = new Main();  
ob.sort(arr, 0, arr.length-1);  
System.out.println("\nSorted array");  
printArray(arr);  
}  
}
```

OUTPUT:



```
Given Array  
12 11 13 5 6 7  
  
Sorted array  
5 6 7 11 12 13  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```


Q10. Write a java program to search for an element in an array and if found print the element and print “Element found.” else print “Element not found”.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int i,n,search,flag=0;

        System.out.println("Enter the number of elements:") ;

        n = sc.nextInt();

        int[] a = new int[n];

        System.out.println("Enter the elements") ;

        for(i=0;i<n;i++)

            a[i] = sc.nextInt();

        System.out.println("Enter the element to be seached");

        search = sc.nextInt();

        for(i=0;i<n;i++) {

            if(a[i]==search)

            {

                System.out.println("Element "+search+" found at "+i+" position");

                flag=1;

                break;

            }

        }

        if(flag==0) {

            System.out.println("Element "+search+" not found");

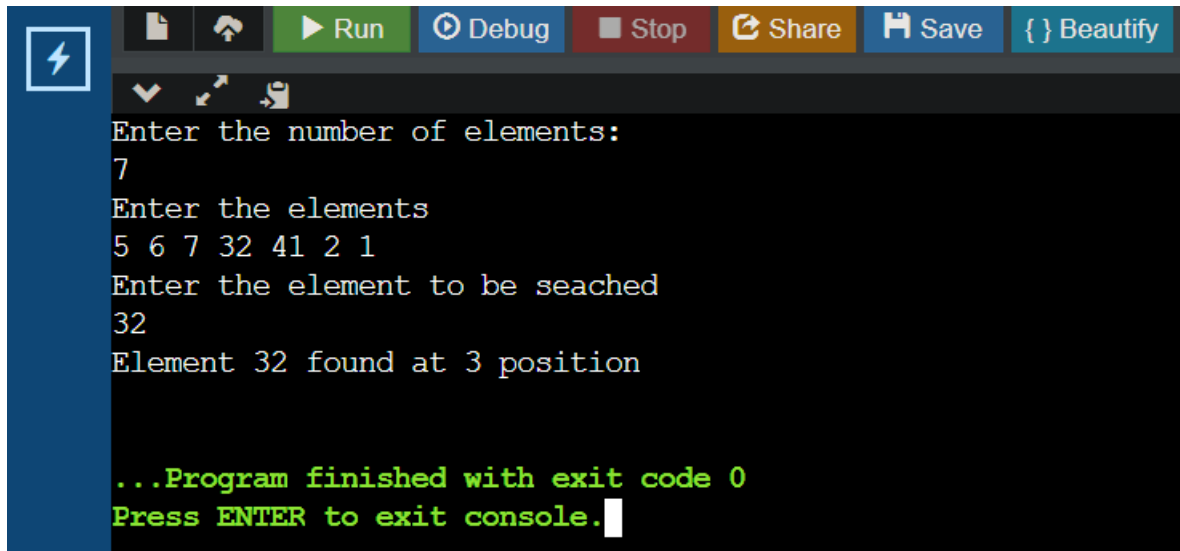
        }

    }

}
```

```
}  
}
```

OUTPUT:



The screenshot shows a code editor's console window with a dark background and light green text. The console displays the following sequence of text: "Enter the number of elements:", "7", "Enter the elements", "5 6 7 32 41 2 1", "Enter the element to be seached", "32", "Element 32 found at 3 position", "...Program finished with exit code 0", and "Press ENTER to exit console." with a white cursor. The editor's toolbar at the top includes buttons for "Run", "Debug", "Stop", "Share", "Save", and "Beautify".

```
Enter the number of elements:  
7  
Enter the elements  
5 6 7 32 41 2 1  
Enter the element to be seached  
32  
Element 32 found at 3 position  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Q11. Write a program in java to rearrange elements in array in such a way that negative number comes before positive number.

CODE:

```
import java.util.*;

public class Main {

    static void rearrangeArray(int arr[], int N) {

        int i, ind = -1;

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

            if (arr[i] == 0) {

                ind = i;

                break;

            }

        }

        int j = -1, k, temp;

        for (k = 0; k < N; k++) {

            if (arr[k] < 0) {

                j += 1;

                if (arr[j] == 0)

                    j += 1;

                temp = arr[j];

                arr[j] = arr[k];

                arr[k] = temp;

            }

        }

        int temp2 = arr[j];

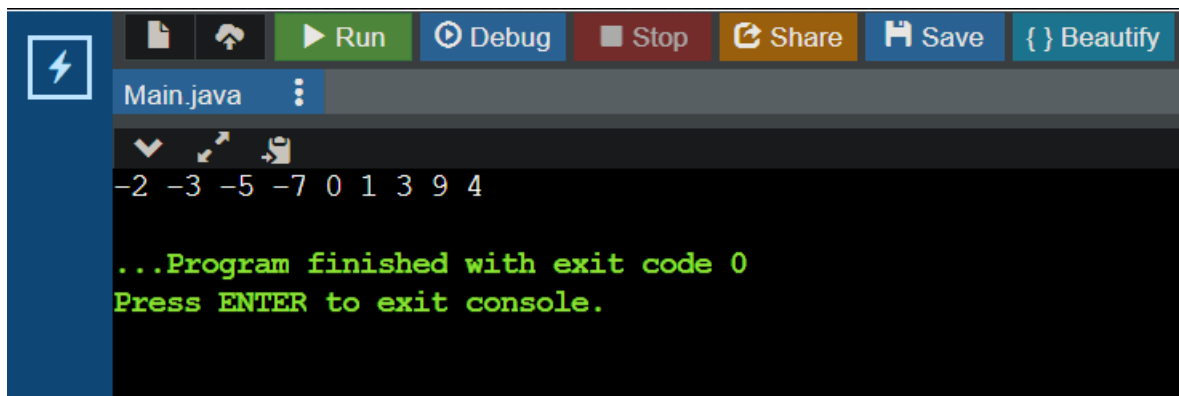
        arr[j] = arr[ind];

        arr[ind] = temp2;

        for (j = 0; j < N; j++) {
```

```
        System.out.print(arr[j] + " ");  
    }  
}  
public static void main (String[] args) {  
    int arr[] = { 1, 0, -2, 3, 4, -5, -7, 9, -3 };  
    int N = arr.length;  
    rearrangeArray(arr, N);  
}  
}
```

OUTPUT:



```
Main.java  
-2 -3 -5 -7 0 1 3 9 4  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Q12. Write a java program to find saddle point coordinates of a matrix.

CODE:

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        System.out.print("Enter the size of 2d matrix : ");

        int n = s.nextInt();

        int arr[][] = new int[n][n];

        System.out.println("enter the array:-");

        for (int i = 0; i < n; i++)

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

                arr[i][j] = s.nextInt();

            }

        int col_max = 0;

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

            int row_min = arr[i][0];

            int col = 0;

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

                if (arr[i][j] < row_min) {

                    row_min = arr[i][j];

                    col = j;

                }

            }

            for (int k = 0; k < n; k++) {

                if (row_min < arr[k][col]) {

                    col_max = 0;

                }

            }

        }

    }

}
```

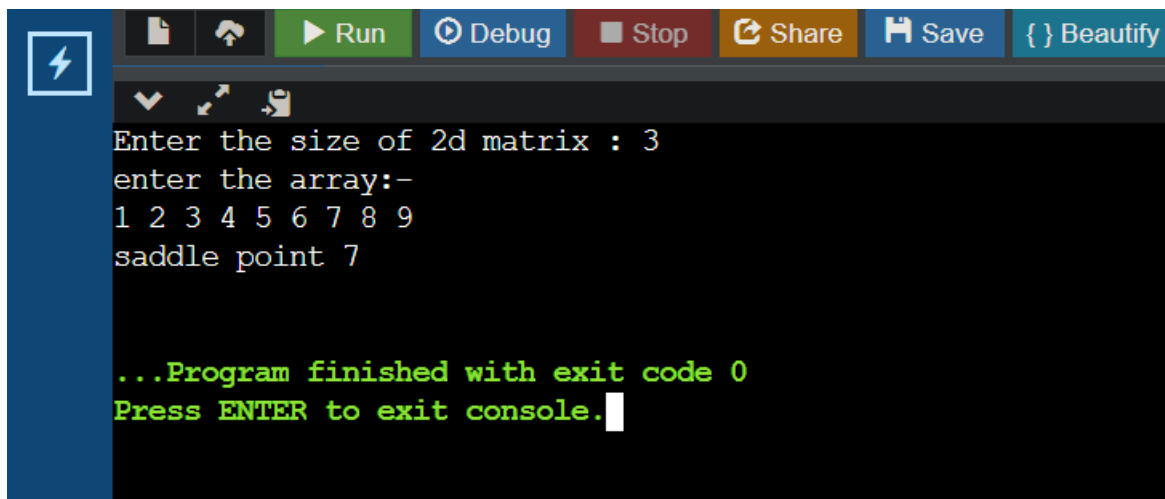
```

        break;
    }
    else
        col_max = row_min;
    }
    if (col_max != 0)
        System.out.println("saddle point " + col_max);
    }

}
}

```

OUTPUT:



The screenshot shows an IDE interface with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, and Beautify. Below the toolbar is a console window with a dark background and light green text. The console output shows the program's execution: it prompts for the size of a 2D matrix (3), then for the array elements (1 2 3 4 5 6 7 8 9), and finally outputs "saddle point 7". The program ends with the message "...Program finished with exit code 0" and a prompt "Press ENTER to exit console." with a cursor.

```

Enter the size of 2d matrix : 3
enter the array:-
1 2 3 4 5 6 7 8 9
saddle point 7

...Program finished with exit code 0
Press ENTER to exit console.

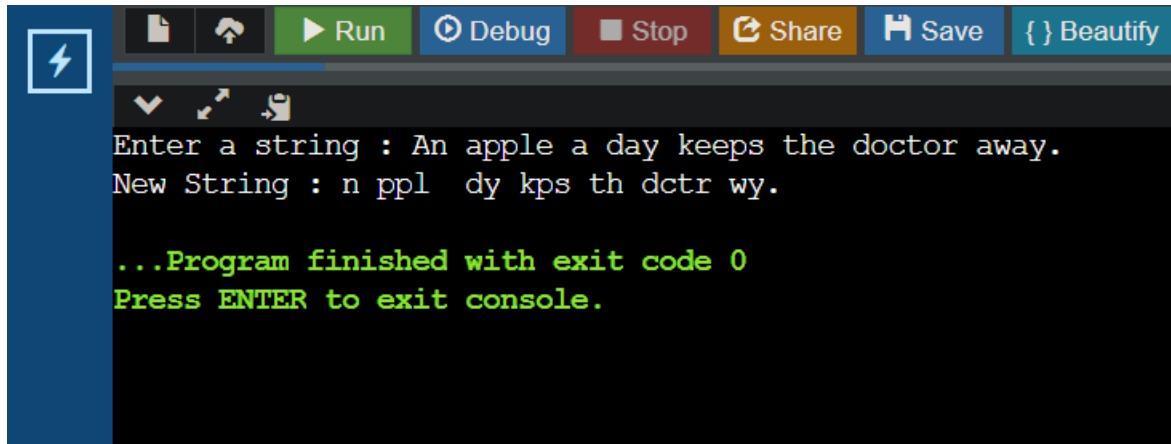
```

Q13. Implement a java program to delete vowels from given string using StringBuffer class.

CODE:

```
import java.util.*;
import java.lang.*;
public class Main {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string : ");
        String str = sc.nextLine();
        StringBuffer sb = new StringBuffer(str);
        for(int i = 0 ; i < sb.length() ; i++) {
            if(sb.charAt(i) == 'a' || sb.charAt(i) == 'e' || sb.charAt(i) == 'i' ||
            sb.charAt(i) == 'o' || sb.charAt(i) == 'u' || sb.charAt(i) == 'A' ||
            sb.charAt(i) == 'E' || sb.charAt(i) == 'T' || sb.charAt(i) == 'O' || sb.charAt(i)
            == 'U') {
                sb.replace(i,i+1,"");
                i--;
            }
        }
        System.out.print("New String : "+ sb.toString());
    }
}
```

OUTPUT:



The screenshot shows a code editor interface with a dark theme. At the top, there is a toolbar with icons for file operations and buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Beautify'. Below the toolbar, the console output is displayed in a monospaced font. The text shows a program that prompts for a string, processes it, and then finishes with an exit code of 0. The output is as follows:

```
Enter a string : An apple a day keeps the doctor away.  
New String : n ppl dy kps th dctr wy.  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```


Q14. WAP to print all the substrings of “O(1+)O”.

CODE:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        String s = "o(1+)o";

        String p = "";

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

            System.out.print(s.charAt(i));

            System.out.print("\n");

            p = "" + s.charAt(i);

            for(int j = i+1 ; j < s.length() ; j++){

                p = p + s.charAt(j);

                System.out.print(p);

                System.out.print("\n");

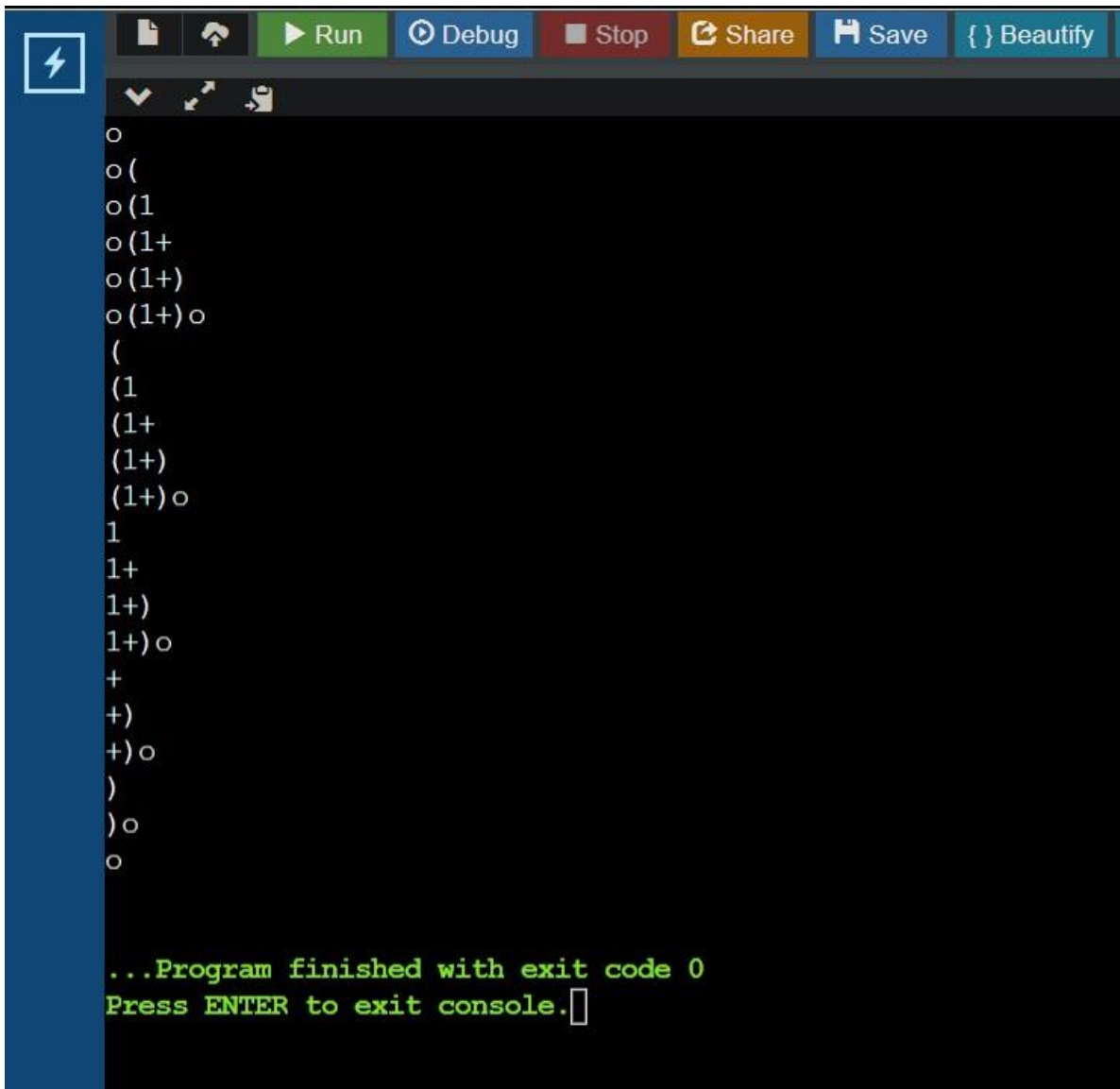
            }

        }

    }

}
```

OUTPUT:



```
o
o(
o(1
o(1+
o(1+)
o(1+)o
(
(1
(1+
(1+)
(1+)o
1
1+
1+)
1+)o
+
+)
+)o
)
)o
o

...Program finished with exit code 0
Press ENTER to exit console.
```