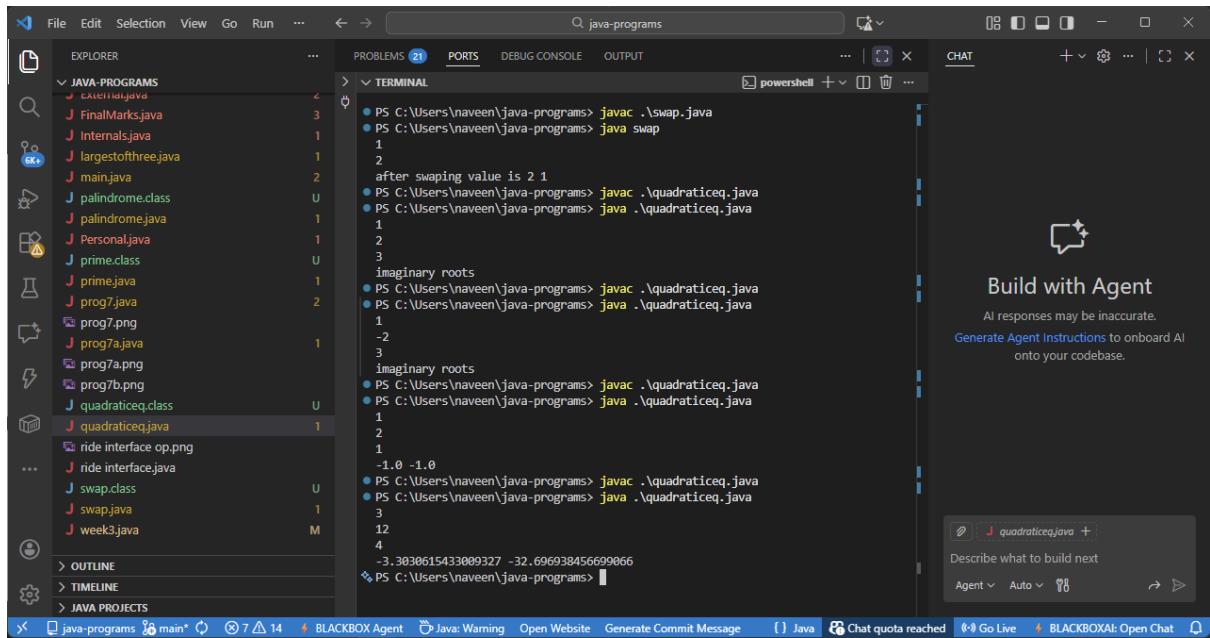


QUADRATIC EQUATION

```
import java.util.*;  
  
public class quadraticeq {  
    public static void main(String[] args) {  
        Scanner sc=new Scanner(System.in);  
        double a=sc.nextInt();  
        double b=sc.nextInt();  
        double c=sc.nextInt();  
        double dis=Math.pow(b,2)-(4*a*c);  
        double r1,r2;  
  
        if(dis>0){  
            r1=(-b+Math.sqrt(dis))/2*a;  
            r2=(-b-Math.sqrt(dis))/2*a;  
            System.out.println(r1+" "+r2);  
        }  
        if(dis==0){  
            r1=-b/2*a;  
            r2=r1;  
            System.out.println(r1+" "+r2);  
        }  
        if(dis<0){  
            System.out.println("imaginary roots");  
        }  
    }  
}
```



1.Calculator

```
import java.util.*;  
  
public class calculator {  
  
    public static void main(String[] args) {  
  
        Scanner s=new Scanner(System.in);  
  
        System.out.println("enter two values of a numbers ");  
  
        int a=s.nextInt();  
  
        int b=s.nextInt();  
  
        System.out.println("enter the option 1-add 2-sub 3-mul 4-div");  
  
        int opt=s.nextInt();  
  
        switch(opt){  
  
            case 1:  
  
                System.out.println(a+b);  
        }  
    }  
}
```

```
break;

case 2:
    System.out.println(a-b);
    break;case 3:
    System.out.println(a*b);
    break;case 4:
    System.out.println(a/b);
    break;
default:
    System.out.println("invalid input");
}
}
}
```

The screenshot shows a Java code editor with a dark theme. The code is a simple calculator program named 'calculator.java'. The code uses a switch statement to perform addition, subtraction, multiplication, or division based on user input. The terminal window below shows the execution of the program, including the prompt for two numbers and the option to choose an operation.

```
J calculator.java 1 X
J calculator.java
2 public class calculator {
3     public static void main(String[] args) {
4         System.out.println("enter two values of a numbers ");
5         int a=s.nextInt();
6         int b=s.nextInt();
7         System.out.println("enter the option 1-add 2-sub 3-mul 4-div");
8         int opt=s.nextInt();
9         switch(opt){
10             case 1:
11                 System.out.println(a+b);
12                 break;
13             case 2:
14                 System.out.println(a-b);
15                 break;case 3:
16                 System.out.println(a*b);
17                 break;case 4:
18                 System.out.println(a/b);
19                 break;
20             }
}
PROBLEMS 21 TERMINAL DEBUG CONSOLE PORTS OUTPUT
● PS C:\Users\naveen\java-programs> javac calculator.java
● PS C:\Users\naveen\java-programs> java calculator
enter two values of a numbers
10
20
enter the option 1-add 2-sub 3-mul 4-div
3
200
```

2.EVEN OR ODD

```
import java.util.*;
public class evenorodd {
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        if(n%2==0)System.out.println("even");
        else System.out.println("odd");
    }
}
```

```
...umberGuessing.java | evenorodd.class | evenorodd.java | X | D
J evenorodd.java > evenorodd > main(String[])
U 1 import java.util.*;
1 2 public class evenorodd {
Run | Debug
1 3     public static void main(String[] args) {
2 4         Scanner s=new Scanner(System.in);
2 5         int n=s.nextInt();
3 6         if(n%2==0)System.out.println(x: "even");
1 7         else System.out.println(x: "odd");
1 8     }
1 9 }
2 10
U
1
1
U
1
2
```

PROBLEMS 21 PORTS DEBUG CONSOLE OUTPUT

> ▾ TERMINAL power

```
PS C:\Users\naveen\java-programs> .\evenorodd.java
● PS C:\Users\naveen\java-programs> java evenorodd
2
even
● PS C:\Users\naveen\java-programs> .\evenorodd.java
● PS C:\Users\naveen\java-programs> java evenorodd
3
odd
❖ PS C:\Users\naveen\java-programs> █
```

3.PALINDROME

```
import java.util.*;

public class palindrome {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        String s=sc.nextLine();

        s=s.toLowerCase();

        int l=0;

        int r=s.length()-1;

        boolean found=true;

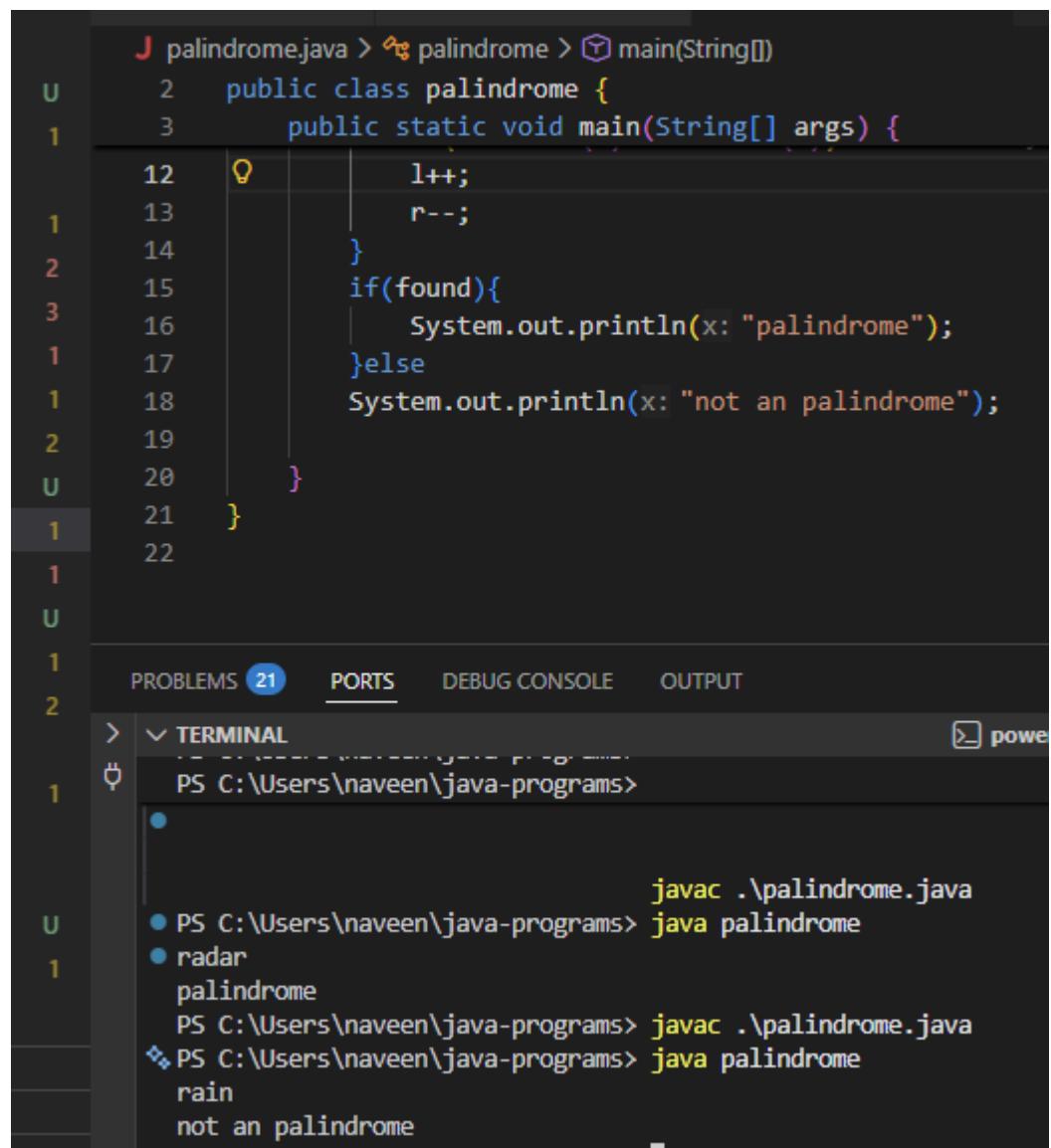
        while(l<=r){
```

```

        if(s.charAt(l)!=s.charAt(r)) found=false;
        l++;
        r--;
    }
    if(found){
        System.out.println("palindrome");
    }else
        System.out.println("not an palindrome");

}
}

```



The screenshot shows a Java code editor with the following code:

```

J palindrome.java > palindrome > main(String[])
U 2     public class palindrome {
1     3         public static void main(String[] args) {
12     4             l++;
13     5             r--;
14     6         }
15     7         if(found){
16             8             System.out.println(x: "palindrome");
17         }else
18             9             System.out.println(x: "not an palindrome");
20     10        }
21     11    }
22

```

The code is annotated with line numbers (1, 2, 3, etc.) and color-coded syntax highlighting. The editor interface includes tabs for PROBLEMS (21), PORTS, DEBUG CONSOLE, and OUTPUT.

In the TERMINAL tab, the following command-line session is shown:

```

PS C:\Users\naveen\java-programs> javac .\palindrome.java
PS C:\Users\naveen\java-programs> java palindrome
radar
palindrome
PS C:\Users\naveen\java-programs> javac .\palindrome.java
PS C:\Users\naveen\java-programs> java palindrome
rain
not an palindrome

```

4.PRIME NUMBER

```
import java.util.*;  
  
public class prime {  
  
    public static boolean isprime(int n){  
  
        if(n<=1) return false;  
  
        for(int i=2;i*i<=n;i++){  
  
            if(n%i==0) return false;  
  
        }  
        return true;  
    }  
  
    public static void main(String[] args) {  
  
        Scanner s=new Scanner(System.in);  
  
        int a=s.nextInt();  
        boolean found=false;  
        for(int i=2;i<=a/2;i++){  
            if(isprime(i) && isprime(a-i)){  
                System.out.println(a+"can be expressed a sum of two prime"+i+" "+(a-i));  
                found=true;  
            }  
        }  
        if(!found){  
            System.out.println("cant be expressed");  
        }  
    }  
}
```

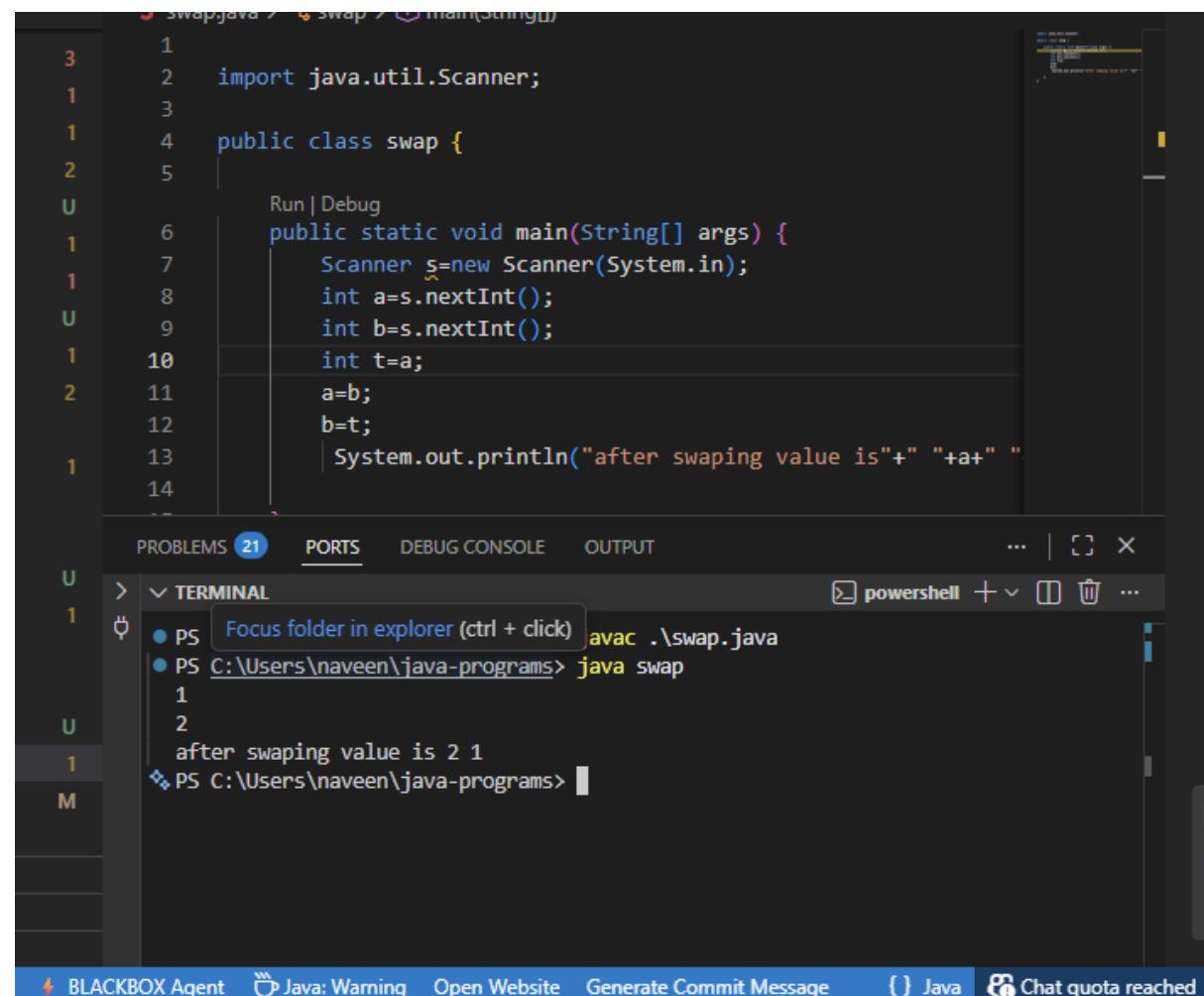
The screenshot shows a Java code editor interface with a dark theme. At the top, there is a code editor window displaying a Java program. The code defines a class named 'prime' with a static method 'isprime' that checks if a number is prime. It also contains a 'main' method that reads an integer from standard input, iterates through possible pairs of prime numbers to find if they sum up to the input, and prints the result. Below the code editor is a navigation bar with tabs: PROBLEMS (21), PORTS, DEBUG CONSOLE, and OUTPUT. The OUTPUT tab is selected, showing the terminal window below. The terminal window displays the execution of the program via the command 'java prime'. The output shows several examples where the input number can be expressed as a sum of two primes, such as 30 = 7 + 23, 30 = 11 + 19, and 30 = 13 + 17. It also shows cases where the input cannot be expressed as a sum of two primes, such as 13 = 13 + 0 and 2 = 0 + 2.

```
0 2 public class prime {  
1 3     public static boolean isprime(int n){  
0 4         if(n==0) return false;  
7  
1 8         return true;  
2 9     }  
3 Run | Debug  
10 public static void main(String[] args) {  
11     Scanner s=new Scanner(System.in);  
12  
13     int a=s.nextInt();  
14     boolean found=false;  
15     for(int i=2;i<=a/2;i++){  
16         if(isprime(i) && isprime(a-i)){  
17             System.out.println(a+" can be expressed as a sum  
18             found=true;  
1  
PROBLEMS 21 PORTS DEBUG CONSOLE OUTPUT ... | ⌂  
1 > ✓ TERMINAL powershell + ⌂ T  
1 PS C:\Users\naveen\java-programs> java prime  
30 can be expressed as a sum of two prime7 23  
30 can be expressed as a sum of two prime11 19  
30 can be expressed as a sum of two prime13 17  
U ● PS C:\Users\naveen\java-programs> javac .\prime.java  
● PS C:\Users\naveen\java-programs> java prime  
13  
13 can be expressed as a sum of two prime2 11  
● PS C:\Users\naveen\java-programs> javac .\prime.java  
● PS C:\Users\naveen\java-programs> java prime  
2  
cant be expressed
```

5.SWAP

```
import java.util.Scanner;  
  
public class swap {  
  
    public static void main(String[] args) {  
        Scanner s=new Scanner(System.in);  
        int a=s.nextInt();  
        int b=s.nextInt();  
        int t=a;  
        a=b;
```

```
b=t;  
System.out.println("after swaping value is"+ " "+a+" "+b);  
  
}  
}
```



The screenshot shows a Java code editor with a dark theme. A Java file named `swap.java` is open, containing the following code:

```
1  
2 import java.util.Scanner;  
3  
4 public class swap {  
5  
    public static void main(String[] args) {  
        Scanner s=new Scanner(System.in);  
        int a=s.nextInt();  
        int b=s.nextInt();  
        int t=a;  
        a=b;  
        b=t;  
        System.out.println("after swaping value is"+ " "+a+" "+b);  
    }  
}
```

Below the code editor is a terminal window titled "TERMINAL". It shows the following session:

```
PS Focus folder in explorer (ctrl + click) avac .\swap.java  
PS C:\Users\naveen\java-programs> java swap  
1  
2  
after swaping value is 2 1  
PS C:\Users\naveen\java-programs>
```

The terminal window has tabs for PROBLEMS (21), PORTS, DEBUG CONSOLE, and OUTPUT. The PROBLEMS tab is active. At the bottom of the interface, there are several status indicators: BLACKBOX Agent, Java: Warning, Open Website, Generate Commit Message, Java, and Chat quota reached.