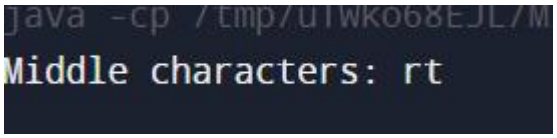
1. public class MiddleCharacter {

```
    public static void main(String[] args) {
        String input = "OpenAIs";
        displayMiddleCharacter(input);
    }
    public static void displayMiddleCharacter(String str) {
        int length = str.length();
        int middleIndex = length / 2;
        if (length % 2 == 0) {
            System.out.println("Middle character: " + str.charAt(middleIndex));
        } else {
            System.out.println("Middle characters: " + str.charAt(middleIndex - 1) +
str.charAt(middleIndex));
        }
    }
}
```

Output:

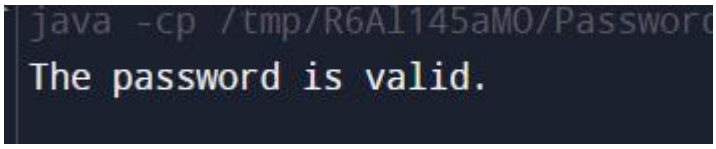



2. public class PasswordValidator {

```
    public static void main(String[] args) {
        String password = "Password1234";
        boolean isValid = isValidPassword(password);
        if (isValid) {
            System.out.println("The password is valid.");
        } else {
            System.out.println("The password is invalid.");
        }
    }
    public static boolean isValidPassword(String password) {
        if (password.length() < 10) {
```

```
            return false;
        }
        int digitCount = 0;
        for (int i = 0; i < password.length(); i++) {
            char ch = password.charAt(i);
            if (!Character.isLetterOrDigit(ch)) {
                return false;
            }
            if (Character.isDigit(ch)) {
                digitCount++;
            }
        }
        return digitCount >= 2;
    }
}
```

Output:




3.
```
import java.util.*;
public class thing{
    public void  ascendingorder(int[] arr){
        Arrays.sort(arr);

    }

public static void main(String[] args){
    int[] arr={10,20,6,4};
    thing Thing=new thing();
    Thing.ascendingorder(arr);
```

```
    System.out.println(Arrays.toString(arr));
}


}
```

Output:

```
java -cp /tmp/tgCooJo8Ax/thing
[4, 6, 10, 20]
```

4.
```
import java.util.*;
public class dog{
    String colour;
    String name;
public dog(String colour,String name){
     this.colour=colour;
    this.name=name;
}
public static void main(String[] args){
    dog s=new dog("black","nani");
    System.out.println("dog colour is" + s.colour);
    System.out.println("dog name is" + s.name);
}
}
```

Output:

```
java -cp /tmp/SZ7muomKnD/dog
dog name isblack
dog colour isnani
```

5.
```
public class book{
    String author;
    String title;
    int price;
```

```
public book(){
}
public book(String author,String title){
   this.author=author;
   this.title=title;
}
public book(String author,String title,int price){
   this.author=author;
   this.title=title;
   this.price=price;
}
public static void main(String[] args){
   book s=new book("chari","life is gone");
   System.out.println("author name is " + s.author);
   System.out.println("title name is " + s.title);
   book n=new book("teja","full damage",13000);
   System.out.println("author name is " + n.author);
   System.out.println("title name is " + n.title);
   System.out.println("price is " + n.price);
}
}
```

Output:

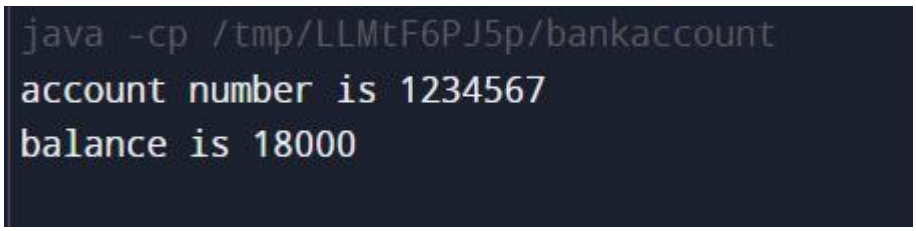
```
java -cp /tmp/hM027aLwS8/book
author name is chari
title name is life is gone
author name is teja
title name is full damage
price is 13000
```

6.

```
public class bankaccount{
   private int accountnumber;
```

```
    private int balance;
public bankaccount(int accountnumber,int balance){
    this.accountnumber=accountnumber;
    this.balance=balance;
}
void setAccountnumber(int accountnumber){
    this.accountnumber=accountnumber;
}
int getAccountnumber(){
    return accountnumber;
}
void setBalance(int balance){
    this.balance=balance;
}
int getBalance(){
    return balance;
}
public static void main(String[] args){
    bankaccount s=new bankaccount(1234567,18000);
    System.out.println("account number is " + s.getAccountnumber());
    System.out.println("balance is " + s.getBalance());
}
}
```

Output:




7.

```
interface playable{
    public void play();
```

```
}
 class football implements playable{
   public void play(){
      System.out.println("this team was played by 11 members");

   }
}
 class volleyball implements playable{
   public void play(){
      System.out.println("only six members are played");
   }
}
 class basketball implements playable{
   public void play(){
      System.out.println("only eight numbers are played");
   }
}
public class person{
public static void main(String[] args){
   football n=new football();
   n.play();
   volleyball b=new volleyball();
   b.play();
   basketball s=new basketball();
   s.play();
}
}
```

Output:

```
java -cp /tmp/APO3iKpa6c/person
this team was played by 11 members
only six members are played
only eight numbers are played
```

8. public class Initializer {

```
    static int initialValue;
    static {
        initialValue = 1000;
        System.out.println("Static block executed. Initial value set to: " + initialValue);
    }
    public Initializer() {
        System.out.println("Initializer instance created.");
    }
    public static void main(String[] args) {
        System.out.println("Value of initialValue before instance creation: " + Initializer.initialValue);
        Initializer initializer = new Initializer();
        System.out.println("Value of initialValue after instance creation: " + Initializer.initialValue);
    }
}
```

Output:

```
java -cp /tmp/kudsa74KHt/Initializer
Static block executed. Initial value set to: 1000
Value of initialValue before instance creation: 1000
Initializer instance created.
Value of initialValue after instance creation: 1000
```

9. public class IDGenerator {

```
    private static int nextID = 1;
    public static int generateID() {
        int currentID = nextID;
        nextID++;
```

```
        return currentID;
    }
    public static void main(String[] args) {
        int id1 = IDGenerator.generateID();
        int id2 = IDGenerator.generateID();
        int id3 = IDGenerator.generateID();
        System.out.println("Generated ID 1: " + id1);
        System.out.println("Generated ID 2: " + id2);
        System.out.println("Generated ID 3: " + id3);
    }
}
```

Output:

```
java -cp /tmp/XuqcNKBoQc/IDGenerator
Generated ID 1: 1
Generated ID 2: 2
Generated ID 3: 3
```

10.

```
import java.util.Scanner;


class VowelException extends Exception {
    public VowelException(String message) {
        super(message);
    }
}
public class vowlesChecker {
    public static void checkvowels(String input) throws VowelException{
        if(!input.matches(".[AEIOUaeiou].")){
        throw new VowelException("the string does not contain vowels.");
    }
    else{
        System.out.println("the string contains vowels.");
    }
```

```
}
   public static void main(String[] args) {
      Scanner scan=new Scanner(System.in);
      String s=scan.nextLine();
      try{
         checkvowels(s);
      }
      catch(VowelException e){
         System.err.println(e.getMessage());
      }
      scan.close();

   }

}
```

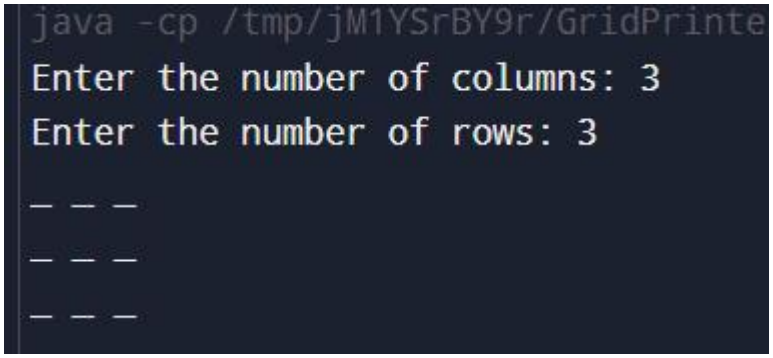
Output:

```
java -cp /tmp/4ta17J2yXE/vowlesChecker
hhhhhhh
the string does not contain vowels.
```

11.
```
import java.util.Scanner;
class GridPrinter{
   public static void main(String[] args){
      Scanner scan = new Scanner(System.in);
      System.out.print("Enter the number of columns: ");
      int cols = scan.nextInt();
      System.out.print("Enter the number of rows: ");
      int rows = scan.nextInt();
      for (int i = 0; i < rows; i++) {
         for (int j = 0; j < cols; j++) {
            System.out.print("_ ");
         }
```

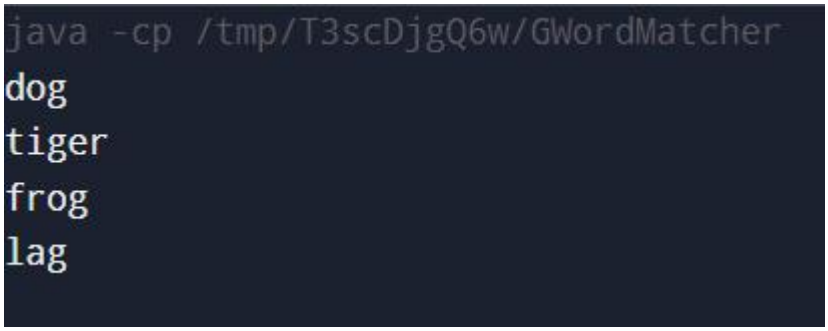
```
            System.out.println();
        }
        scan.close();
    }
}
```

Output:




12.
```
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class GWordMatcher {
    public static void main(String[] args) {
        String inputString = "dog cat tiger lion frog lag";
        Pattern pattern = Pattern.compile("\\b\\w*g\\w*\\b");
        Matcher matcher = pattern.matcher(inputString);
        while (matcher.find()) {
            String matchedWord = matcher.group();
            System.out.println(matchedWord);
        }
    }
}
```

Output:




13. import java.util.ArrayList;

```
import java.util.List;
public class ListMerger {
   public static <T> List<T> mergeLists(List<T> list1, List<T> list2) {
      List<T> mergedList = new ArrayList<>();
      int size1 = list1.size();
      int size2 = list2.size();
      int maxSize = Math.max(size1, size2);
      for (int i = 0; i < maxSize; i++) {
         if (i < size1) {
            mergedList.add(list1.get(i));
         }
         if (i < size2) {
            mergedList.add(list2.get(i));
         }
      }
      return mergedList;
   }
   public static void main(String[] args) {
      List<Integer> list1 = new ArrayList<>();
      list1.add(1);
      list1.add(3);
      list1.add(5);
      List<Integer> list2 = new ArrayList<>();
      list2.add(2);
```

```
        list2.add(4);
        list2.add(6);
        list2.add(8);
        List<Integer> mergedList = mergeLists(list1, list2);
        System.out.println("Merged List: " + mergedList);
        List<String> list3 = new ArrayList<>();
        list3.add("A");
        list3.add("C");
        list3.add("E");
        List<String> list4 = new ArrayList<>();
        list4.add("B");
        list4.add("D");
        list4.add("F");
        list4.add("G");
        List<String> mergedListStrings = mergeLists(list3, list4);
        System.out.println("Merged String List: " + mergedListStrings);
    }
}
```

Output:

```
java -cp /tmp/XUdxazp4DN/ListMerger
Merged List: [1, 2, 3, 4, 5, 6, 8]
Merged String List: [A, B, C, D, E, F, G]
```

14.
```
public class selectionsort {
    public static void sortarray(int[] arr){
        int n=arr.length;
        for(int i=0;i<n;i++){
            int min_index=i;
            for (int j=i+1;j<n;j++){
                if(arr[j]<arr[min_index]){
                    min_index=j;
```

```
            }
        }
        int temp=arr[min_index];
        arr[min_index]=arr[i];
        arr[i]=temp;
    }
  }
  public static void main(String[] args) {
    int[] arr={4,5,6,2,1,3};
    sortarray(arr);
    printarray(arr);
  }
    public static void printarray(int[] arr){
    for(int num:arr){
      System.out.print(num+" ");
    }
    System.out.println();
    }
}
```

Output:

```
java -cp /tmp/ICg1myJ3dn/selectionsort
1 2 3 4 5 6
```

```
15. public class search {
  public static void binarysearch(int[] arr,int target){
    int n=arr.length;
    for (int i=0;i<n;i++){
      if(arr[i]==target){
        System.out.println("element found at index : "+i);
      }
    }
```

```
    }
    public static void main(String[] a){
        int[] arr={1,2,3,4,5,6,7,8};
        int target=5;
        binarysearch(arr,target);
    }

}
```

Output:

```
java -cp /tmp/Q2L4W6ZrYU/search
element found at index : 4
```

16.
```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class UnderscoreSequences {
    public static void main(String[] args) {
        String inputString = "abc_def_ghi_123_xyz";
        Pattern pattern = Pattern.compile("[a-z]+_[a-z]+");
        Matcher matcher = pattern.matcher(inputString);

        while (matcher.find()) {
            String sequence = matcher.group();
            System.out.println(sequence);
        }
    }
}
```

Output:

```
java -cp /tmp/hgDcMmRZlF/UnderscoreSequer
abc_def
```

17.
```
class OddNumberException extends Exception {
```

```
    public OddNumberException(String message) {
        super(message);
    }
}
public class NumberChecker {
    public static void checkEven(int number) throws OddNumberException {
        if (number % 2 != 0) {
            throw new OddNumberException("The number " + number + " is odd. Exception thrown.");
        }
        System.out.println("The number " + number + " is even.");
    }
    public static void main(String[] args) {
        try {
            checkEven(10);
            checkEven(11);
        } catch (OddNumberException e) {
            System.out.println("Exception caught: " + e.getMessage());
            System.out.println(e);
        }
    }
}
```

Output: