

Question 01.

```

package Question01;
public class Question01 {
    public static void main(String[] args) {
        Stack stackS = new Stack(5);
        char wordArray[] = {'a','b','c','d','e'};
        for (char a: wordArray) {
            stackS.push(a);
        }
        System.out.println("");
        while (!stackS.isEmpty()){
            System.out.println(stackS.pop());
        }
    }
}

```

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'DSA-LAB-01' with a source folder 'src' containing files 'Question01', 'Stack', and 'Test'.
- Editor:** Displays the code for 'Stack.java'. It includes a private array 'charArr' and a 'push' method that adds elements to the array and increments a 'top' pointer. The 'pop' method is also visible, removing elements from the array and decrementing 'top'.
- Run Console:** Shows the output of the program. It first prints an empty string, then inserts elements 'a' through 'e' into the stack. Subsequently, it retrieves and prints the values 'e', 'd', 'c', 'b', and 'a' in reverse order, demonstrating the LIFO (Last In, First Out) behavior of a stack. The process finishes with exit code 0.

```
package Question01;

public class Stack {
    private int top;
    private int maxSize;
    private char[] charArr;

    public Stack(int size) {
        this.maxSize = size;
        charArr = new char[maxSize];
        top=-1;
    }

    public boolean isStackEmpty(){
        return top== -1;
    }

    public boolean isStackFull(){
        return top==maxSize-1;
    }

    public int getTop() {
        return top;
    }

    public void push(char newValue){
        if (isStackFull()){
            System.out.println("Stack Already Full");
        }
        else {
            System.out.println("Inserting "+newValue);
            charArr[++top] = newValue;
        }
    }

    public char pop(){
        if (isStackEmpty()){
            System.out.println("Stack is Already Empty");
            return charArr[top];
        }
        else {
            System.out.println("Retrieving Value");
            return charArr[top--];
        }
    }
}
```

Question 02.

```

package Question02;

public class Question02 {
    public static void main(String[] args) {
        Stack stackS = new Stack(6);
        String wordArray[] = {"Department", "of", "Statistics", "and",
"Computer", "Science"};
        for (String a : wordArray) {
            stackS.push(a);
        }
        System.out.println("");
        String fOut = "";
        while (!stackS.isEmpty()) {
            fOut += stackS.pop();
            fOut += " ";
            System.out.println("Output : " + fOut);
        }
    }
}

```

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows a project named 'DSA-LAB-01' with a source folder 'src' containing subfolders 'Quest' and 'Stack'. The 'Stack' folder is selected.
- Code Editor:** Displays the implementation of the 'Stack' class.


```

1 usage
public Stack(int size) {
    this.maxSize = size;
    StrArr = new String[maxSize];
    top=-1;
}

2 usages
public boolean isEmpty(){
    return top== -1;
}

```
- Run Console:** Shows the output of the program.


```

Run: Unnamed x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\Je
Inserting Department
Inserting of
Inserting Statistics
Inserting and
Inserting Computer
Inserting Science

Output : Science
Output : Science Computer
Output : Science Computer and
Output : Science Computer and Statistics
Output : Science Computer and Statistics of
Output : Science Computer and Statistics of Department

Process finished with exit code 0

```

```
package Question02;
public class Stack {
    private int top;
    private int maxSize;
    private String[] StrArr;

    public Stack(int size) {
        this.maxSize = size;
        StrArr = new String[maxSize];
        top=-1;
    }

    public boolean isEmpty(){
        return top==-1;
    }

    public boolean isStackFull(){
        return top==maxSize-1;
    }

    public int getTop() {
        return top;
    }

    public void push(String newValue){
        if (isStackFull()){
            System.out.println("Quest.Question02.Stack Already Full");
        }
        else {
            System.out.println("Inserting "+newValue);
            StrArr[++top] = newValue;
        }
    }

    public String pop(){
        if (isEmpty()){
            System.out.println("Quest.Question02.Stack is Already Empty");
            return StrArr[top];
        }
        else {
            return StrArr[top--];
        }
    }
}
```

Question 03.

```
package Question03;

public class Question03 {

    public boolean IsPalindromeOrNot;
    // Write a program to check whether a given string is palindrome using Stack
    // Operations.
    // Example: madam, mom, rotator (Any word that reads the same forward or
    // backward)
    public boolean palindromeChecker(String obj) {
        char[] c = obj.toCharArray();
        Stack stackS = new Stack(c.length);
        for (char a:obj.toCharArray()){
            stackS.push(a);
        }
        System.out.println("Word: "+obj);
        while (!stackS.isEmpty()) {
            for (char a : obj.toCharArray()) {
                if (a != stackS.pop()) {
                    IsPalindromeOrNot = false;
                } else {
                    IsPalindromeOrNot = true;
                }
            }
        }

        return IsPalindromeOrNot;
    }
}
```

```
package Question03;
public class Stack {
    private int top;
    private int maxSize;
    private char[] charArr;

    public Stack(int size) {
        this.maxSize = size;
        charArr = new char[maxSize];
        top=-1;
    }

    public boolean isStackEmpty(){
        return top== -1;
    }

    public boolean isStackFull(){
        return top==maxSize-1;
    }

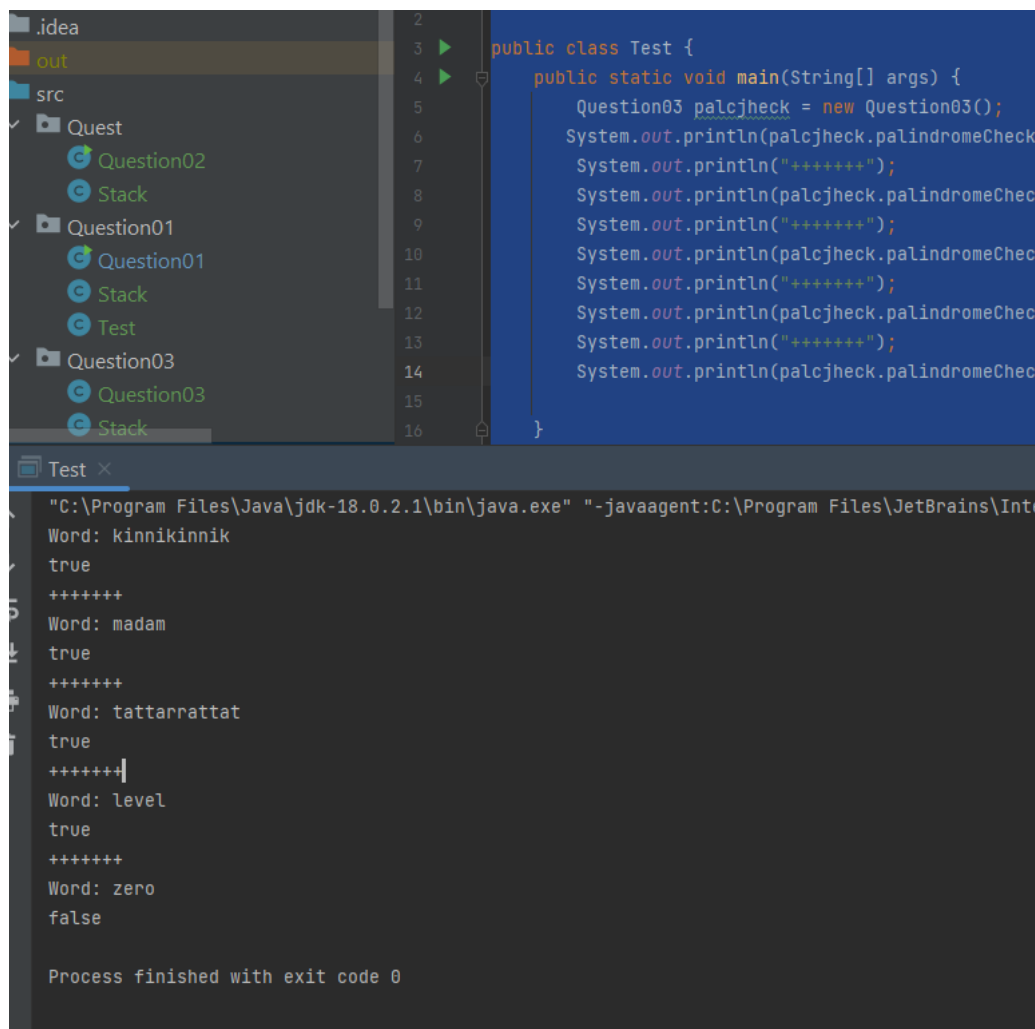
    public int getTop() {
        return top;
    }

    public void push(char newValue){
        if (isStackFull()){
            System.out.println("Stack Already Full");
        }
        else {
            // System.out.println("Inserting "+newValue);
            charArr[++top] = newValue;
        }
    }

    public char pop(){
        if (isStackEmpty()){
            System.out.println("Stack is Already Empty");
            return charArr[top];
        }
        else {
            // System.out.println("Retrieving Value");
            return charArr[top--];
        }
    }
}
```

```
package Question03;

public class Test {
    public static void main(String[] args) {
        Question03 palcjcheck = new Question03();
        System.out.println(palcjcheck.palindromeChecker("kinnikinnik"));
        System.out.println("+++++++");
        System.out.println(palcjcheck.palindromeChecker("madam"));
        System.out.println("+++++++");
        System.out.println(palcjcheck.palindromeChecker("tattarrattat"));
        System.out.println("+++++++");
        System.out.println(palcjcheck.palindromeChecker("level"));
        System.out.println("+++++++");
        System.out.println(palcjcheck.palindromeChecker("zero"));
    }
}
```



Question 04.

```
package Question04;
public class Stack {
    private int top;
    private int maxSize;
    private char[] charArr;

    public Stack(int size) {
        this.maxSize = size;
        charArr = new char[maxSize];
        top = -1;
    }

    public boolean isEmpty() {
        return top == -1;
    }

    public boolean isFull() {
        return top == maxSize - 1;
    }

    public int getTop() {
        return top;
    }

    public void push(char newValue) {
        if (isFull()) {
            System.out.println("Stack Already Full");
        } else {
            // System.out.println("Inserting "+newValue);
            charArr[++top] = newValue;
        }
    }

    @Override
    public String toString() {
        System.out.println(charArr);
        return charArr.toString();
    }

    public char pop() {
        if (isEmpty()) {
            System.out.println("Stack is Already Empty");
            return charArr[top];
        } else {
            // System.out.println("Retrieving Value");
            return charArr[top--];
        }
    }
}
```



```
package Question04;

import java.util.ArrayList;
import java.util.Collection;

public class Question04 {

    int aCount=0;
    int eCount=0;
    int iCount=0;
    int oCount=0;
    int uCount=0;

    public void VowelChecker(String inpString){

        Stack stackForWord = new Stack(inpString.length());
        Stack stackForVowel = new Stack(inpString.length());
        char[] inpStringArray = inpString.toCharArray();
        for (char c : inpStringArray) {
            stackForWord.push(c);
        }

        ArrayList<Character> charArrayList = new ArrayList<>();
        for (char c : inpStringArray) {
            if (c=='a' && aCount == 0) {
                charArrayList.add(c);
                aCount++;
            } else if(c=='e'&& eCount==0){
                charArrayList.add(c);
                eCount++;
            } else if(c=='i'&& iCount==0){
                charArrayList.add(c);
                iCount++;
            } else if(c=='o'&& oCount==0){
                charArrayList.add(c);
                oCount++;
            }
        }
    }
}
```

```
        } else if(c=='u' && uCount==0){
            charArrayList.add(c);
            uCount++;
        }
    }
    java.util.Collections.sort(charArrayList);

    for (Character character : charArrayList.reversed()) {
        stackForVowel.push(character);
    }
    while (!stackForVowel.isEmpty()) {
        System.out.printf(stackForVowel.pop()+"");
    }
    System.out.println();
}
}
```

```
package Question04;
```

```
public class Test {
    public static void main(String[] args) {
        Question04 example01 = new Question04();
        Question04 example02 = new Question04();
        Question04 example03 = new Question04();
        Question04 example04 = new Question04();
        Question04 example05 = new Question04();
        Question04 example06 = new Question04();
        Question04 example07 = new Question04();
        Question04 example08 = new Question04();
        Question04 example09 = new Question04();
        Question04 example10 = new Question04();
    }
}
```

```
Question04 example11 = new Question04();
```

```
Question04 example12 = new Question04();
```

```
example01.VowelChecker("apple");    // Example 1
```

```
example02.VowelChecker("zebra");    // Example 2
```

```
example03.VowelChecker("school");    // Example 3
```

```
example04.VowelChecker("banana");    // Example 4
```

```
example05.VowelChecker("library");    // Example 5
```

```
example06.VowelChecker("elephant");  // Example 6
```

```
example07.VowelChecker("orange");    // Example 7
```

```
example08.VowelChecker("unicorn");   // Example 8
```

```
example09.VowelChecker("grape");     // Example 9
```

```
example10.VowelChecker("python");    // Example 10
```

```
example11.VowelChecker("kiwi");      // Example 11
```

```
example12.VowelChecker("jasmine");   // Example 12
```

```
}
```

```
}
```

The screenshot shows a code editor with a dark theme. The top bar has tabs for 'Problems 1', 'Output', 'Debug Console', and 'Terminal'. The 'Terminal' tab is active, showing the output of a Java program. The output lists vowels: 'ae', 'o', 'i', 'aei'. Below this, a command prompt shows the execution of a Java program. The command is: `cd /Users/kavindus/BECS-21223-Data-Structures-and-Algorithms-LAB-01-1 ; /usr/bin/env /Library/Java/JavaVirtualMachines/temurin-24.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/kavindus/Library/Application\ Support/Trae/User/workspaceStorage/0af89945fe396e0331538c850fbb081d/redhat.java/jdt_ws/BECS-21223-Data-Structures-and-Algorithms-LAB-01-1_db8972b7/bin Question04.Test`. The output of the program is: `ae`, `ae`, `o`, `a`, `ai`, `ae`, `aeo`, `iou`, `ae`, `o`. The status bar at the bottom indicates 'Format on save: Disabled', 'Not Committed Yet', and 'Ln 31, Col 5 (1188 selected)'.

```
ae
o
i
aei
kavindus@kavindus-MacBook-Air BECS-21223-Data-Structures-and-Algorithms-LAB-01-1 % cd /Users
/kavindus/BECS-21223-Data-Structures-and-Algorithms-LAB-01-1 ; /usr/bin/env /Library/Java/Jav
aVirtualMachines/temurin-24.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsI
nExceptionMessages -cp /Users/kavindus/Library/Application\ Support/Trae/User/workspaceStorag
e/0af89945fe396e0331538c850fbb081d/redhat.java/jdt_ws/BECS-21223-Data-Structures-and-Algorith
ms-LAB-01-1_db8972b7/bin Question04.Test
ae
ae
o
a
ai
ae
aeo
iou
ae
o
```

Question 05.

```
package Question05;
public class Stack {
    private int top;
    private int maxSize;
    private String[] StrArr;

    public Stack(int size) {
        this.maxSize = size;
        StrArr = new String[maxSize];
        top=-1;
    }

    public boolean isEmpty(){
        return top==-1;
    }

    public boolean isStackFull(){
        return top==maxSize-1;
    }

    public int getTop() {
        return top;
    }

    public void push(int newValue){
        if (isStackFull()){
            System.out.println("Quest.Question02.Stack Already Full");
        }
        else {
            StrArr[++top] = String.valueOf(newValue);
        }
    }

    public String pop(){
        if (isEmpty()){
            System.out.println("Quest.Question02.Stack is Already Empty");
            return StrArr[top];
        }
        else {
            // System.out.println("Retrieving Value");
            return StrArr[top--];
        }
    }
}
```

```
package Question05;
public class StackString {
    private int top;
    private int maxSize;
    private String[] StrArr;

    public StackString(int size) {
        this.maxSize = size;
        StrArr = new String[maxSize];
        top=-1;
    }

    public boolean isStackEmpty(){
        return top==-1;
    }

    public boolean isStackFull(){
        return top==maxSize-1;
    }

    public int getTop() {
        return top;
    }

    public void push(String newValue){
        if (isStackFull()){
            System.out.println("Quest.Question02.Stack Already Full");
        }
        else {
            StrArr[++top] = String.valueOf(newValue);
        }
    }

    public String pop(){
        if (isStackEmpty()){
            System.out.println("Quest.Question02.Stack is Already Empty");
            return StrArr[top];
        }
        else {
            // System.out.println("Retrieving Value");
            return StrArr[top--];
        }
    }

}
```

```
package Question05;
public class Question05 {
    Stack stackforBin = new Stack(32);
    Stack stackforOctal = new Stack(18);
    StackString stack = new StackString(12);

    public void binaryConvert(int no){
        while(!(no==0)){
            stackforBin.push((char) (no%2));
            no /=2;
        }
        while (!stackforBin.isStackFull()){
            stackforBin.push(0);
        }
        while (!stackforBin.isStackEmpty()){
            System.out.printf(stackforBin.pop()+"");
        }
        System.out.println();
    }

    public void octalConvert(int no){
        while(!(no==0)){
            stackforOctal.push((char) (no%8));
            no /=8;
        }
        while (!stackforOctal.isStackFull()){
            stackforOctal.push(0);
        }
        while (!stackforOctal.isStackEmpty()){
            System.out.printf(stackforOctal.pop()+"");
        }
        System.out.println();
    }

    public void hexaConvert(int no) {
        while (!(no == 0)) {
            if (no % 16 == 10) {
                stack.push("A");
                no /= 16;
            } else if (no % 16 == 11) {
                stack.push("B");
                no /= 16;
            } else if (no % 16 == 12) {
                stack.push("C");
                no /= 16;
            } else if (no % 16 == 13) {
                stack.push("D");
            }
        }
    }
}
```

```

        no /= 16;
    } else if (no % 16 == 14) {
        stack.push("E");
        no /= 16;
    } else if (no % 16 == 15) {
        stack.push("F");
        no /= 16;
    } else {
        stack.push(String.valueOf(no % 16));
        no /= 16;
    }
}

while (!stack.isStackFull()) {
    stack.push(String.valueOf(0));
}
while (!stack.isStackEmpty()) {
    System.out.printf(stack.pop() + "");
}
System.out.println();

}
}

```

```
package Question05;

public class Test {
    public static void main(String[] args) {
        Question05 question05 = new Question05();
        question05.binaryConvert(77);
        question05.octalConvert(77);
        question05.hexaConvert(77);
    }
}
```

[illegible]