2) import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 while (true) {  
 int user;  
 System.*out*.println("enter the youre flight no '1,2,3'");  
 user = sc.nextInt();  
 if(user>0 && user<4){  
 System.*out*.println("good");  
 break;  
 }  
 else{  
 System.*out*.println("invalid");  
 }  
 }  
 }  
 }

4. public class Main{  
 public static void main(String[] args){  
 int n = 10, firstno = 0, secondno = 1;  
 int nestno;  
 for(int i = 1; i<=n; i++){  
 System.*out*.print(firstno + ",");  
 nestno = firstno + secondno;  
 firstno = secondno;  
 secondno = nestno;  
  
 }  
 }  
}

5. public class Main{  
 public static void main(String[] args){  
 int arr [][] = {  
 {1,2},  
 {4,5},  
  
 };  
 for(int i=0;i<2;i++){  
 for(int j=0;j<2;j++){  
 System.*out*.print(arr[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
}

14. public class Main{  
 public static int find\_max(int[] arr){  
 int temp = arr[0];  
 for(int i = 0;i<arr.length;i++){  
 if(arr[i]>temp){  
 temp = arr[i];  
 }  
 }  
 return temp;  
 }  
  
 public static void main(String[] args){  
 int numbers[] = {8,2,4,1,3,5,6};  
 System.*out*.println("biggest number is " + *find\_max*(numbers));  
 }  
}

19. public class Main {  
 public static void main(String[] args) {  
 int arr\_1[] = {1,2,3,4,5};  
 int arr\_2 [] = {6,7,8,9};  
 int merge\_arr [] = new int[arr\_1.length +arr\_2.length];  
 int i =0,j=0,k=0;  
 while(i<arr\_1.length && j<arr\_2.length){  
 if(arr\_1[i]<arr\_2[j]){  
 merge\_arr[k] = arr\_1[i];  
 i++;  
 }  
 else{  
 merge\_arr[k] = arr\_2[j];  
 j++;  
 }  
 k++;  
 }  
 while (i<arr\_1.length){  
 merge\_arr[k] = arr\_1[i];  
 i++;  
 k++;  
 }  
 while (j<arr\_2.length){  
 merge\_arr[k] = arr\_2[j];  
 j++;  
 k++;  
 }  
 for(int l = 0;l< merge\_arr.length;l++){  
 System.*out*.println(merge\_arr[l]);  
 }  
 }  
}

20. public class Main {  
 public static void reversed(int[] arr){  
 int start = 0;  
 int end = arr.length-1;  
 while (start<end){  
 int temp = arr[start];  
 arr[start] =arr[end];  
 arr[end] = temp;  
 start++;  
 end--;  
  
 }  
 }  
  
 public static void main(String[] args) {  
 int arr\_1[] = {1,2,3,4,5};  
 *reversed*(arr\_1);  
 for(int i=0;i<arr\_1.length;i++){  
 System.*out*.println(arr\_1[i]);  
 }  
 }  
}

21. import java.util.Arrays;  
public class Main {  
 public static void main(String[] args) {  
 int arr\_1[] = {4,2,3,1};  
 Arrays.*sort*(arr\_1);  
 System.*out*.println("second largest number "+arr\_1[arr\_1.length-2]);  
 }  
}

31public class Main {  
 public static void main(String[] args) {  
 String str = "aeiouftg";  
 String words[] = str.split(",");  
 int sum = 0;  
 for (int i=0;i<str.length();i++){  
 char c = str.charAt(i);  
 if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u'){  
 sum+=1;  
 }  
 }  
 System.*out*.println(sum);  
  
 }  
}

32. public class Main {  
 public static void main(String[] args) {  
 int row = 5;  
 for(int i = row;i>0;i--){  
 for(int j=0;j<i;j++){  
 System.*out*.print("1 ");  
 }  
 System.*out*.println(" ");  
 }  
 }  
}

33. import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int times = 0;  
 int total = 0;  
 while (true) {  
 System.*out*.println("Enter rating from 1 to 5");  
 int user = sc.nextInt();  
 if(user!=0){  
 times++;  
 total+=user;  
 }  
 else{  
 break;  
 }  
 }  
 int avg = total/times;  
 System.*out*.println("the average rating is " + avg);  
 System.*out*.println("the total rating is" + times);  
 }  
}

34. import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int times = 0;  
 int total = 0;  
 while (true) {  
 System.*out*.println("Enter expenses for food, transportation etc.");  
 String user = sc.nextLine();  
 if (!user.equals("done")) {  
 try{  
 int expense = Integer.*parseInt*(user);  
 total += expense;  
 }catch (NumberFormatException e){  
 System.*out*.println("please enter a valid number");  
 }  
 }  
 else{  
 break;  
 }  
 }  
 System.*out*.println("The total expense is "+total);  
 }  
}

34. Develop a password validation system that prompts users to create a password. The program should check if the password meets certain criteria (length, special characters, etc.). If it doesn't meet the criteria, it should continue prompting the user until a valid password is entered.

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 while(true){  
 System.*out*.println("please enter a valid password");  
 String str = sc.nextLine();  
 if(*is\_correct*(str)){  
 System.*out*.println("Successfull");  
 break;  
 }  
 else{  
 System.*out*.println("envalid try again");  
 }  
 }  
  
 }  
  
 public static boolean is\_correct(String pass){  
 if(pass.length()<8) return false;  
 boolean haslower = false;  
 boolean hasupper = false;  
 boolean hasdigit = false;  
 boolean hassymbol = false;  
  
 for(char c : pass.toCharArray()){  
 if(Character.*isLowerCase*(c)) haslower = true;  
 else if (Character.*isUpperCase*(c)) hasupper = true;  
 else if (Character.*isDigit*(c)) hasdigit =true;  
 else if (!Character.*isLetterOrDigit*(c)) hassymbol = true;  
 }  
 return hasdigit && haslower && hasupper && hassymbol;  
 }  
}

37. Develop a temperature conversion tool that allows users to convert temperatures between Celsius and Fahrenheit. Use a loop to continue asking for temperature values until the user chooses to exit. After each conversion, display the result and prompt the user again.

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 while(true){  
 System.*out*.println("enter rempratue in celsius");  
 String celsius = sc.nextLine();  
  
 if(celsius.equals("done")){  
 break;  
 }  
 else{  
 try{  
 double cel = Double.*parseDouble*(celsius);  
 double faren = (cel\*9/5)+32;  
 System.*out*.println("in faren" + faren);  
 }catch(NumberFormatException e){  
 System.*out*.println("error occured");  
 }  
 }  
 }  
 }  
}

38. mplement a simple banking system where users can deposit and withdraw money. Use a loop to allow the user to perform transactions until they choose to exit. After exiting, display the final account balance and transaction history.

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int balance = 12000;  
 List<String> transactions = new ArrayList<>();  
  
 while(true){  
 System.*out*.println("choose operation withdraw or deposit");  
 String user = sc.nextLine();  
 if(user.equals("withdraw")){  
 System.*out*.println("Enter ammount");  
 int ammount = sc.nextInt();  
 if(ammount<balance){  
 System.*out*.println("successfull");  
 balance-=ammount;  
 System.*out*.println("balance" + balance );  
 transactions.add("Withdraw" + ammount);  
 }  
 else{  
 System.*out*.println("unsufficient balance");  
 }  
  
 } else if (user.equals("deposit")) {  
 System.*out*.println("Enter ammount");  
 int depo\_ammount = sc.nextInt();  
 balance+=depo\_ammount;  
 System.*out*.println("successful");  
 transactions.add("deposit" + depo\_ammount);  
 System.*out*.println("Total ammount "+balance);  
   
 } else if (user.equals("history")) {  
 for(String transaction : transactions){  
 System.*out*.println(transaction);  
 }  
 } else if (user.equals("exit")) {  
 break;  
 }  
 }  
  
 }  
}

38. Create a program that allows a teacher to input grades for students in a class. The program should continue to prompt for grades until the teacher enters -1 to stop. After all grades have been entered, calculate and display the average grade, the highest grade, and the number of students who passed (e.g., scored above a certain threshold). share code

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 List<Integer> grades = new ArrayList<>();  
 int highest\_score = 0;  
 int pass\_limit = 30;  
 int passing\_count = 0;  
 int total = 0;  
 while(true){  
 System.*out*.println("Enter a grade or -1 for exit");  
 int marks = sc.nextInt();  
 if(marks==-1){  
 break;  
 }  
 if(marks>0 && marks<=100){  
 total += marks;  
 grades.add(marks);  
 if(marks>pass\_limit){  
 passing\_count++;  
 }  
 }  
 if(marks>highest\_score){  
 highest\_score=marks;  
 }  
  
 }  
 double avg = total/grades.size();  
 System.*out*.println("avg marks are "+avg);  
 System.*out*.println("students passed "+passing\_count);  
 System.*out*.println("highest acore "+ highest\_score);  
  
 }  
}

41. Write a program that calculates the total sales and commission for a group of salespeople. Prompt the user to enter sales figures for each salesperson in a loop. The loop should continue until a negative number is entered, indicating the end of input. Calculate and display the total sales and the average sales per salesperson.

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int sales =0;  
 int sale\_person =0;  
 while (true){  
 System.*out*.println("Enter sale ammount or -1 to exit");  
 int user = sc.nextInt();  
 if(user<0){  
 break;  
 }  
 else{  
 sales+=user;  
 sale\_person++;  
 }  
 }  
 double avg = sales/sale\_person;  
 System.*out*.println("total sales are "+sales);  
 System.*out*.println("avg sales are "+avg);  
 }  
}

42. Write a Java program to reverse a String.

import java.util.\*;  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 String str = "hello";  
 String reverse = "";  
 for(int i =str.length()-1 ; i>=0;i--){  
 reverse+=str.charAt(i);  
 }  
 System.*out*.println(reverse);  
 }  
}

47. Write a program that counts the number of vowels and consonants in a given String. Input: "Hello World" Output: Vowels: 3, Consonants: 7

public class VowelConsonantCounter {

public static void main(String[] args) {

String input = "Hello World"; // Input string

int vowels = 0; // Variable to store the count of vowels

int consonants = 0; // Variable to store the count of consonants

// Convert the input string to lowercase to handle both uppercase and lowercase letters

input = input.toLowerCase();

// Loop through each character in the string

for (char c : input.toCharArray()) {

// Check if the character is a vowel

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

vowels++;

}

// Check if the character is a consonant (ignoring spaces and non-alphabetic characters)

else if (c >= 'a' && c <= 'z') {

consonants++;

}

}

// Output the results

System.out.println("Vowels: " + vowels);

System.out.println("Consonants: " + consonants);

}

}

48. Implement a method that capitalizes the first letter of each word in a given String. Input: "hello world" Output: "Hello World"

public class CapitalizeWords {

public static void main(String[] args) {

String input = "hello world"; // Input string

String result = capitalizeFirstLetter(input); // Call method to capitalize words

System.out.println(result); // Output the result

}

// Method to capitalize the first letter of each word

public static String capitalizeFirstLetter(String input) {

// Split the string into words using space as a delimiter

String[] words = input.split(" ");

StringBuilder result = new StringBuilder(); // StringBuilder to build the result

// Loop through each word

for (String word : words) {

// Capitalize the first letter and keep the rest of the word as is

if (word.length() > 0) {

result.append(Character.toUpperCase(word.charAt(0))) // Capitalize first letter

.append(word.substring(1)) // Append the rest of the word

.append(" "); // Add space after each word

}

}

// Return the result string with trimmed space at the end

return result.toString().trim();

}

}

49. Implement a method that checks if two Strings are anagrams of each other (contain the same characters in a different order). Input: "listen", "silent" Output: true

import java.util.Arrays;

public class AnagramChecker {

public static void main(String[] args) {

String str1 = "listen"; // First string

String str2 = "silent"; // Second string

boolean result = areAnagrams(str1, str2); // Call method to check for anagrams

System.out.println(result); // Output the result

}

// Method to check if two strings are anagrams

public static boolean areAnagrams(String str1, String str2) {

// If the lengths of the strings are different, they cannot be anagrams

if (str1.length() != str2.length()) {

return false;

}

// Convert both strings to character arrays

char[] arr1 = str1.toCharArray();

char[] arr2 = str2.toCharArray();

// Sort both arrays

Arrays.sort(arr1);

Arrays.sort(arr2);

// Compare the sorted arrays

return Arrays.equals(arr1, arr2);

}

}

50. Write a program to remove duplicate characters from a String while maintaining the original order of characters. Input: "programming" Output: "progamin"

public class RemoveDuplicates {

public static void main(String[] args) {

String input = "programming"; // Input string

String result = removeDuplicates(input); // Call method to remove duplicates

System.out.println(result); // Output the result

}

// Method to remove duplicate characters while maintaining the original order

public static String removeDuplicates(String input) {

StringBuilder result = new StringBuilder(); // StringBuilder to store result

boolean[] seen = new boolean[256]; // Boolean array to track encountered characters

// Loop through each character of the input string

for (char c : input.toCharArray()) {

if (!seen[c]) { // If character is not encountered yet

result.append(c); // Append to result

seen[c] = true; // Mark character as encountered

}

}

return result.toString(); // Convert StringBuilder to String and return

}

}

51. Create a method that finds the first non-repeating character in a String. Input: "swiss" Output: 'w'

import java.util.HashMap;

public class FirstNonRepeatingCharacter {

public static void main(String[] args) {

String input = "swiss"; // Input string

char result = firstNonRepeatingCharacter(input); // Call method to find first non-repeating character

System.out.println(result); // Output the result

}

// Method to find the first non-repeating character

public static char firstNonRepeatingCharacter(String input) {

HashMap<Character, Integer> charCountMap = new HashMap<>(); // Map to store character counts

// Loop through the string to count occurrences of each character

for (char c : input.toCharArray()) {

charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);

}

// Loop through the string again to find the first character with count 1

for (char c : input.toCharArray()) {

if (charCountMap.get(c) == 1) {

return c; // Return the first non-repeating character

}

}

// If no non-repeating character is found, return a placeholder (e.g., '\0')

return '\0';

}

}

52. Implement a method to compress a String using the counts of repeated characters. If the compressed String is not smaller than the original, return the original String. Input: "aabcccccaaa" Output: "a2b1c5a3"

public class StringCompression {

public static void main(String[] args) {

String input = "aabcccccaaa"; // Input string

String result = compressString(input); // Call the method to compress the string

System.out.println(result); // Output the result

}

// Method to compress a string

public static String compressString(String input) {

// If the input string is empty or has a single character, return it as is

if (input == null || input.length() <= 1) {

return input;

}

StringBuilder compressed = new StringBuilder(); // StringBuilder to store compressed string

int count = 1; // Initialize count for consecutive characters

// Loop through the string starting from the second character

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

if (input.charAt(i) == input.charAt(i - 1)) {

// If the current character is the same as the previous one, increment count

count++;

} else {

// Otherwise, append the previous character and its count to the result

compressed.append(input.charAt(i - 1)).append(count);

count = 1; // Reset count for the next character

}

}

// Append the last character and its count

compressed.append(input.charAt(input.length() - 1)).append(count);

// Return the compressed string only if it's shorter than the original string

return compressed.length() < input.length() ? compressed.toString() : input;

}

}

53. Write a Java program that appends the string " World" to an existing StringBuffer containing "Hello". Input: "Hello"

public class StringBufferExample {

public static void main(String[] args) {

// Create a StringBuffer containing "Hello"

StringBuffer strBuffer = new StringBuffer("Hello");

// Append " World" to the StringBuffer

strBuffer.append(" World");

// Output the result

System.out.println(strBuffer); // Output: "Hello World"

}

}

54. Create a method that inserts the string "Beautiful " at index 6 in the StringBuffer containing "Hello World". Input: "Hello World" Output: "Hello Beautiful World"

public class StringBufferExample {

public static void main(String[] args) {

// Create a StringBuffer containing "Hello"

StringBuffer strBuffer = new StringBuffer("Hello");

// Append " World" to the StringBuffer

strBuffer.append(" World");

// Output the result

System.out.println(strBuffer); // Output: "Hello World"

}

}

55 Write a Java program that reverses the contents of a StringBuffer initialized with "Java Programming". Input: "Java Programming" Output: "gnimmargorPavaJ"

public class StringBufferReverseExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "Java Programming"

StringBuffer strBuffer = new StringBuffer("Java Programming");

// Reverse the StringBuffer

strBuffer.reverse();

// Output the reversed StringBuffer

System.out.println(strBuffer); // Output: "gnimmargorPavaJ"

}

}

56. Create a method that deletes a substring from a StringBuffer. For example, remove "World" from "Hello World". Input: "Hello World" Output: "Hello "

public class StringBufferDeleteExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "Hello World"

StringBuffer strBuffer = new StringBuffer("Hello World");

// Call the method to remove the substring "World"

deleteSubstring(strBuffer, "World");

// Output the modified StringBuffer

System.out.println(strBuffer); // Output: "Hello "

}

// Method to delete a substring from a StringBuffer

public static void deleteSubstring(StringBuffer strBuffer, String substring) {

// Find the start and end indices of the substring

int startIndex = strBuffer.indexOf(substring);

// If the substring is found, delete it

if (startIndex != -1) {

strBuffer.delete(startIndex, startIndex + substring.length());

}

}

}

57. Write a program that initializes a StringBuffer with "Java Programming" and reverses its content. Input: "Java Programming" Output: "gnimmargorP avaJ"

public class StringBufferReverseExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "Java Programming"

StringBuffer strBuffer = new StringBuffer("Java Programming");

// Reverse the StringBuffer

strBuffer.reverse();

// Output the reversed StringBuffer

System.out.println(strBuffer); // Output: "gnimmargorP avaJ"

}

}

58. Create a method that deletes the substring "World" from a StringBuffer initialized with "Hello World". Print the modified StringBuffer. Input: "Hello World" Output: "Hello "

public class StringBufferDeleteExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "Hello World"

StringBuffer strBuffer = new StringBuffer("Hello World");

// Call the method to delete the substring "World"

deleteSubstring(strBuffer, "World");

// Output the modified StringBuffer

System.out.println(strBuffer); // Output: "Hello "

}

// Method to delete a substring from a StringBuffer

public static void deleteSubstring(StringBuffer strBuffer, String substring) {

// Find the start index of the substring

int startIndex = strBuffer.indexOf(substring);

// If the substring is found, delete it

if (startIndex != -1) {

// Delete the substring from startIndex to endIndex (start + length of substring)

strBuffer.delete(startIndex, startIndex + substring.length());

}

}

}

59. Write a Java program that replaces "Java" with "Python" in a StringBuffer initialized with "I love Java programming". Input: "I love Java programming" Output: "I love Python programming"

public class StringBufferReplaceExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "I love Java programming"

StringBuffer strBuffer = new StringBuffer("I love Java programming");

// Replace "Java" with "Python"

int startIndex = strBuffer.indexOf("Java"); // Find the start index of "Java"

if (startIndex != -1) {

strBuffer.replace(startIndex, startIndex + "Java".length(), "Python"); // Replace with "Python"

}

// Output the modified StringBuffer

System.out.println(strBuffer); // Output: "I love Python programming"

}

}

60. Write a program that creates a StringBuffer, checks its initial capacity, and then appends enough characters to exceed that capacity. Print the new capacity. Input: Initial capacity of StringBuffer Output: New capacity after appending characters

public class StringBufferCapacityExample {

public static void main(String[] args) {

// Create a StringBuffer with an initial capacity of 16 (default is 16 if not specified)

StringBuffer strBuffer = new StringBuffer(16);

// Print the initial capacity of the StringBuffer

System.out.println("Initial capacity: " + strBuffer.capacity());

// Append enough characters to exceed the initial capacity

strBuffer.append("This is a long string that exceeds the initial capacity of the StringBuffer.");

// Print the new capacity of the StringBuffer after appending

System.out.println("New capacity after appending: " + strBuffer.capacity());

}

}

61. Write a method that converts a StringBuffer to a String and returns it. Initialize a StringBuffer with "Hello World" and use your method to print the resulting string. Input: StringBuffer initialized with "Hello World" Output: "Hello World"

public class StringBufferToStringExample {

public static void main(String[] args) {

// Initialize a StringBuffer with "Hello World"

StringBuffer strBuffer = new StringBuffer("Hello World");

// Convert StringBuffer to String using the method

String result = convertToString(strBuffer);

// Print the resulting string

System.out.println(result); // Output: "Hello World"

}

// Method to convert StringBuffer to String

public static String convertToString(StringBuffer strBuffer) {

return strBuffer.toString(); // Convert and return the string

}

}

62. Create a method that counts the number of vowels in a StringBuffer. Initialize it with any string and print the number of vowels. Input: "Hello World" Output: 3

public class VowelCountExample {

public static void main(String[] args) {

// Initialize a StringBuffer with the string "Hello World"

StringBuffer strBuffer = new StringBuffer("Hello World");

// Call the method to count vowels and print the result

int vowelCount = countVowels(strBuffer);

// Print the number of vowels

System.out.println("Number of vowels: " + vowelCount); // Output: 3

}

// Method to count the number of vowels in a StringBuffer

public static int countVowels(StringBuffer strBuffer) {

// Convert StringBuffer to String

String str = strBuffer.toString();

int count = 0;

// Loop through each character in the string

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

char c = str.charAt(i); // Get each character

// Check if the character is a vowel (both lowercase and uppercase)

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||

c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {

count++; // Increment the vowel count

}

}

return count; // Return the total count of vowels

}

}

63. Write a Java program that initializes a StringBuffer with extra spaces (e.g., " Hello World ") and trims the whitespace from both ends. Input: " Hello World " Output: "Hello World"

public class TrimStringBufferExample {

public static void main(String[] args) {

// Initialize a StringBuffer with extra spaces

StringBuffer strBuffer = new StringBuffer(" Hello World ");

// Trim the spaces from both ends

strBuffer = new StringBuffer(strBuffer.toString().trim());

// Print the result after trimming

System.out.println("Trimmed String: '" + strBuffer.toString() + "'"); // Output: "Hello World"

}

}

64. Create a method that takes two StringBuffer objects and merges them into one, separating them with a space. Print the resulting StringBuffer. Input: StringBuffer1: "Hello", StringBuffer2: "World" Output: "Hello World"

public class MergeStringBuffers {

public static void main(String[] args) {

// Initialize two StringBuffer objects

StringBuffer strBuffer1 = new StringBuffer("Hello");

StringBuffer strBuffer2 = new StringBuffer("World");

// Call the method to merge the two StringBuffer objects

StringBuffer mergedString = mergeStringBuffers(strBuffer1, strBuffer2);

// Print the merged StringBuffer

System.out.println("Merged String: " + mergedString); // Output: "Hello World"

}

// Method to merge two StringBuffer objects, separated by a space

public static StringBuffer mergeStringBuffers(StringBuffer buffer1, StringBuffer buffer2) {

// Append the second StringBuffer to the first one, with a space in between

buffer1.append(" ").append(buffer2);

// Return the merged StringBuffer

return buffer1;

}

}