JAVA

**1.Reverse a String**:

|  |
| --- |
| public String reverseString(String s) {  char[] charArray = s.toCharArray();  int left = 0, right = charArray.length - 1;  while (left < right) {  char temp = charArray[left];  charArray[left] = charArray[right];  charArray[right] = temp;  left++;  right--;  }  return new String(charArray);  } |

2.FIND MISSING NUMBER:

|  |
| --- |
| import java.util.\*;  class HelloWorld {  public static void main(String[] args) {  int[] arr ={1,2,3,4,6,7,8,9,10};  int reverse = findMiss(arr);  System.out.print(reverse);  }  public static int findMiss(int[] num) {  int n =num.length;  int total=(n+1)\*(n+2)/2;  for(int nums: num) {  total-=nums;  }  return total;  }  } |

**Implement a Stack**:

|  |
| --- |
| public class Stack {  private int maxSize;  private int[] stackArray;  private int top;  public Stack(int size) {  maxSize = size;  stackArray = new int[maxSize];  top = -1;  }  public void push(int value) {  if (top == maxSize - 1) {  System.out.println("Stack is full");  return;  }  stackArray[++top] = value;  }  public int pop() {  if (top == -1) {  System.out.println("Stack is empty");  return -1;  }  return stackArray[top--];  }  public boolean isEmpty() {  return (top == -1);  }  } |

**Find Maximum Subarray Sum**:

|  |
| --- |
| public int maxSubArray(int[] nums) {  int maxSum = nums[0], currentSum = nums[0];  for (int i = 1; i < nums.length; i++) {  currentSum = Math.max(nums[i], currentSum + nums[i]);  maxSum = Math.max(maxSum, currentSum);  }  return maxSum;  } |

1. **Merge Two Sorted Arrays**: