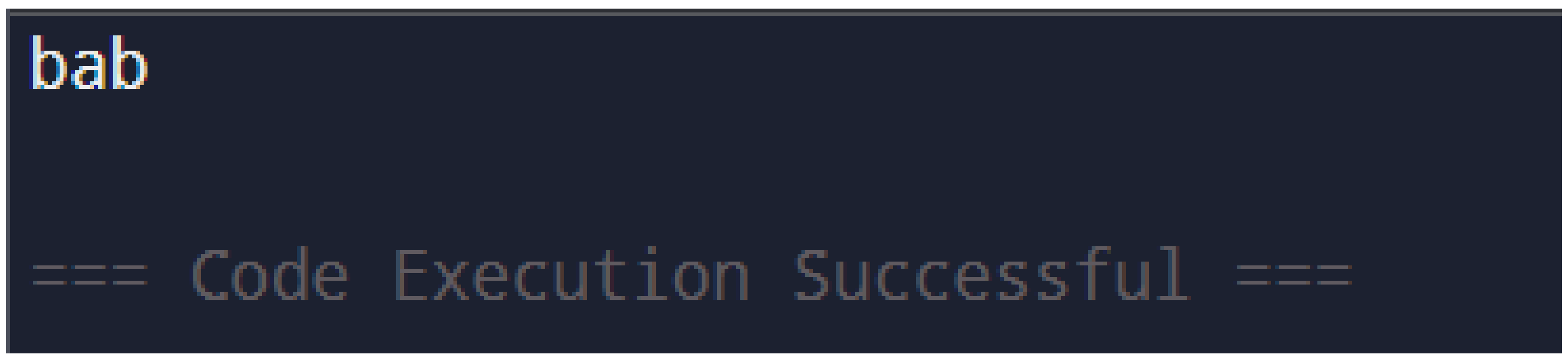
11.A

PROGRAM:

```
public class LongestPalindrome {
    static String longestPal(String s) {
        int n = s.length(), start = 0, max = 1;
        boolean[][] dp = new boolean[n][n];
        for (int i = 0; i < n; i++) dp[i][i] = true;
        for (int len = 2; len <= n; len++)
            for (int i = 0; i < n - len + 1; i++) {
                int j = i + len - 1;
                if (s.charAt(i) == s.charAt(j) && (len == 2 || dp[i+1][j-1])) {
                    dp[i][j] = true;
                    if (len > max) { start = i; max = len; }
                }
            }
        return s.substring(start, start + max);
    }

    public static void main(String[] args) {
        System.out.println(longestPal("babad")); // Output: bab or aba
    }
}
```

OUTPUT:



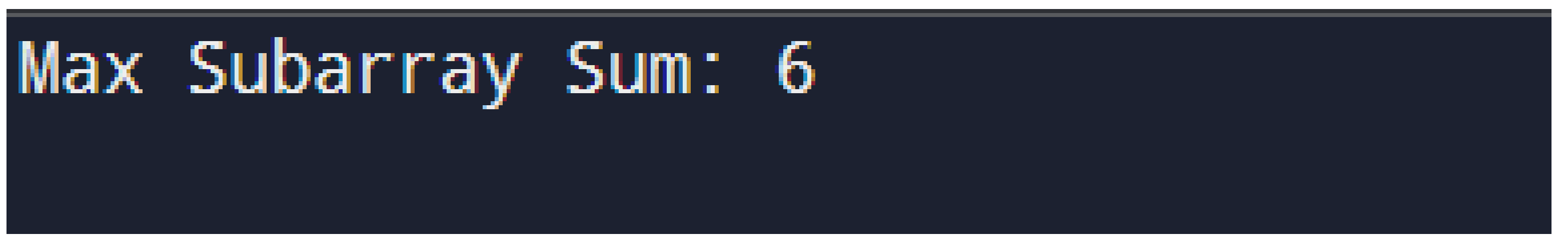
```
bab


=== Code Execution Successful ===
```

11.b

Program:

```
public class MaxSubarray {
    public static void main(String[] args) {
        int[] arr = {-2, 1, -3, 4, -1, 2, 1, -5, 4};
        int maxSum = arr[0], cur = arr[0];
        for (int i = 1; i < arr.length; i++) {
            cur = Math.max(arr[i], cur + arr[i]);
            maxSum = Math.max(maxSum, cur);
        }
        System.out.println("Max Subarray Sum: " + maxSum); // Output: 6
    }
}
```

Output:



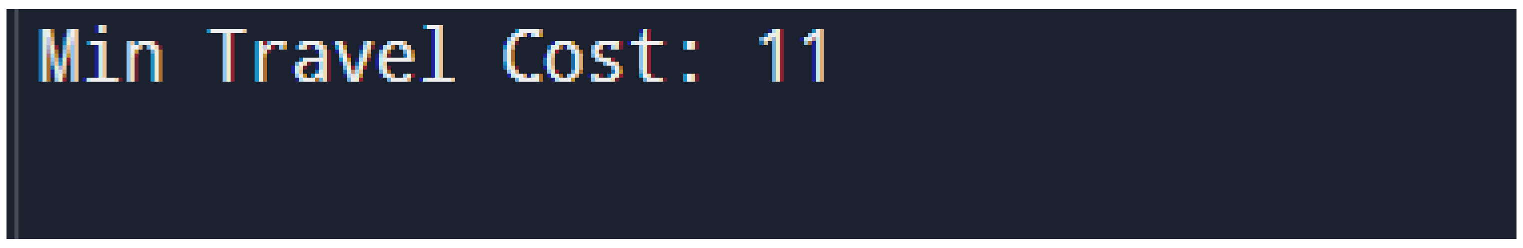
```
Max Subarray Sum: 6
```

11.c

Program:

```
import java.util.*;
public class MinTravelCost {
    public static void main(String[] args) {
        int[] days = {1,4,6,7,8,20}, cost = {2,7,15};
        int last = days[days.length - 1];
        int[] dp = new int[last + 1];
        Set<Integer> travel = new HashSet<>();
        for (int d : days) travel.add(d);

        for (int i = 1; i <= last; i++) {
            if (!travel.contains(i)) dp[i] = dp[i - 1];
            else dp[i] = Math.min(dp[Math.max(0, i - 1)] + cost[0],
                         Math.min(dp[Math.max(0, i - 7)] + cost[1],
                                  dp[Math.max(0, i - 30)] + cost[2]));
        }
        System.out.println("Min Travel Cost: " + dp[last]); // Output: 11
    }
}
```

Output:



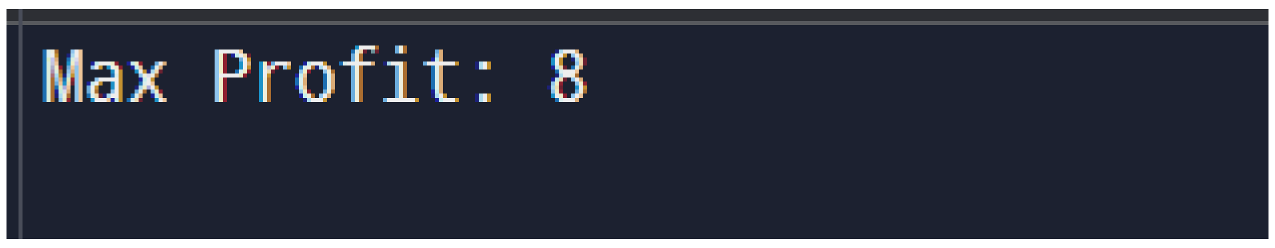
```
Min Travel Cost: 11
```

12.A

Program:

```
public class StockProfit {
  public static void main(String[] args) {
    int[] prices = {1,3,2,8,4,9};
    int fee = 2, hold = -prices[0], cash = 0;
    for (int i = 1; i < prices.length; i++) {
      cash = Math.max(cash, hold + prices[i] - fee);
      hold = Math.max(hold, cash - prices[i]);
    }
    System.out.println("Max Profit: " + cash); // Output: 8
  }
}
```

Output:



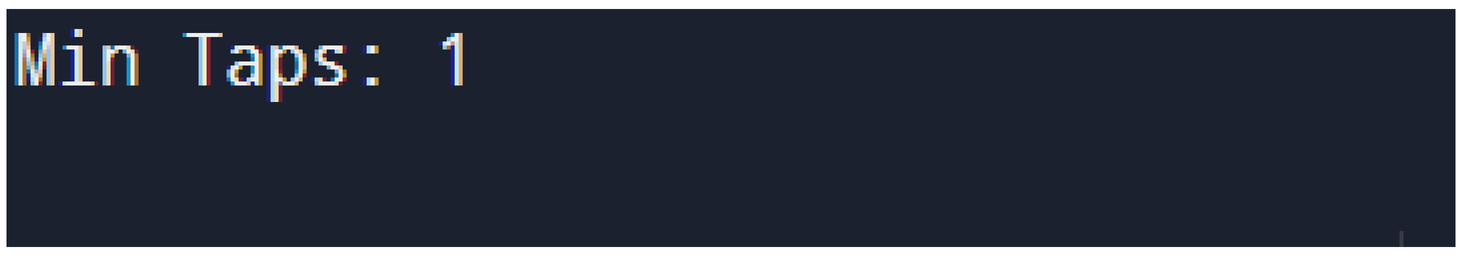
```
Max Profit: 8
```

12.B

Program:

```
import java.util.*;
public class MinTaps {
    public static void main(String[] args) {
        int n = 5; int[] ranges = {3,4,1,1,0,0};
        int[] dp = new int[n + 1];
        Arrays.fill(dp, n + 2);
        dp[0] = 0;
        for (int i = 0; i <= n; i++) {
            int l = Math.max(0, i - ranges[i]), r = Math.min(n, i + ranges[i]);
            for (int j = l; j <= r; j++)
                dp[r] = Math.min(dp[r], dp[l] + 1);
        }
        System.out.println("Min Taps: " + (dp[n] >= n + 2 ? -1 : dp[n]));
    }
}
```

Output:



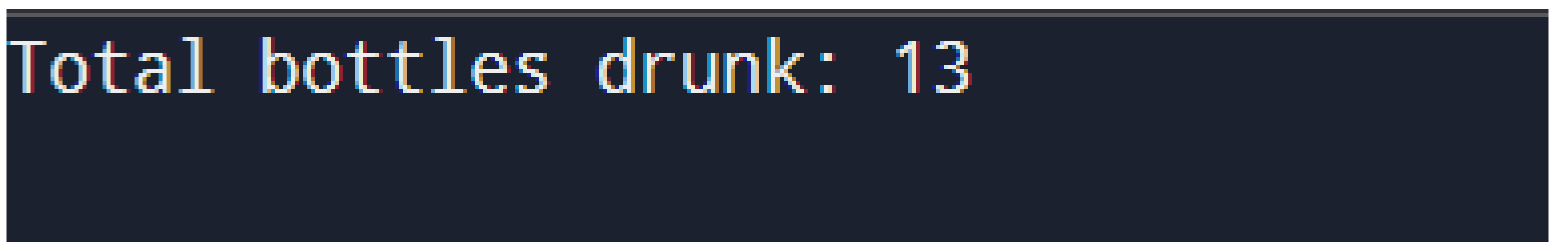
```
Min Taps: 1
```

12.C

Program:

```
public class WaterBottles {
    public static void main(String[] args) {
        int full = 9, exchange = 3, total = 0;
        while (full >= exchange) {
            total += exchange;
            full = full - exchange + 1;
        }
        total += full;
        System.out.println("Total bottles drunk: " + total); // Output: 13
    }
}
```

Output:



```
Total bottles drunk: 13
```