

1. Go program to find duplicate elements in array.

```
package main

import "fmt"

func main() {
    var a []int
    arr := []int{1, 2, 3, 4, 5, 4, 3}
    visited := make(map[int]bool, 0)
    for i := 0; i < len(arr); i++ {
        if visited[arr[i]] == true {
            a = append(a, arr[i])
        } else {
            visited[arr[i]] = true
        }
    }
    fmt.Println(a)
}
```

Output:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\Gowthamvishu\Desktop\golang> go run duplicate.go
[4 3]
PS C:\Users\Gowthamvishu\Desktop\golang> |
```

2. Go program to find largest and smallest number in an array.

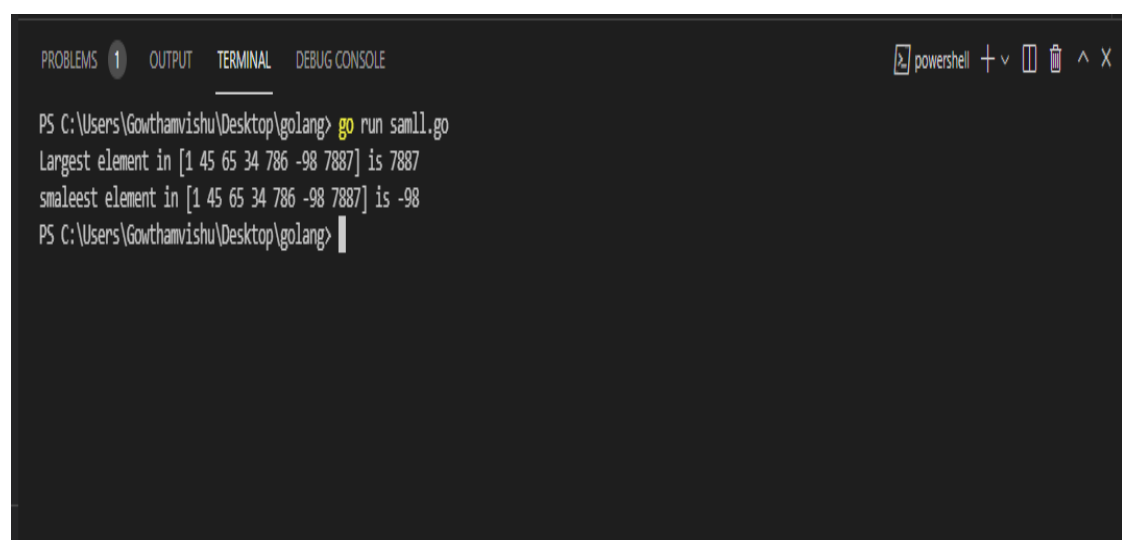
```
package main

import "fmt"

func main() {
    a := []int{1, 45, 65, 34, 786, -98, 7887}
    element := a[0]
    element1 := a[0]
    for _, v := range a {
        if v > element {
            element = v
        }
    }
    for _, u := range a {
        if u < element1 {
            element1 = u
        }
    }

    fmt.Printf("Largest element in %d is %d\n", a, element)
    fmt.Printf("smaleest element in %d is %d", a, element1)
}
```

Output:

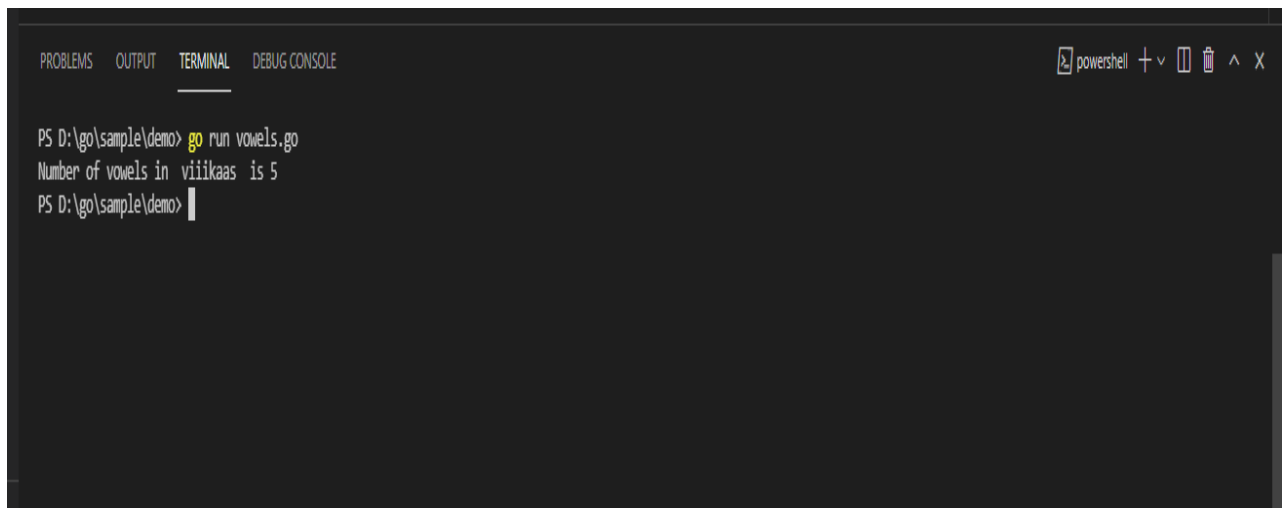
A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', and 'DEBUG CONSOLE'. The 'TERMINAL' tab is active. The prompt is 'PS C:\Users\Gowthamvishu\Desktop\golang>'. The user enters 'go run samll.go'. The output is 'Largest element in [1 45 65 34 786 -98 7887] is 7887' followed by 'smaleest element in [1 45 65 34 786 -98 7887] is -98'. The prompt returns. The window title bar shows 'powershell' and standard window controls.

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\Gowthamvishu\Desktop\golang> go run samll.go
Largest element in [1 45 65 34 786 -98 7887] is 7887
smaleest element in [1 45 65 34 786 -98 7887] is -98
PS C:\Users\Gowthamvishu\Desktop\golang>
```

3. Go program to count number of vowels in a given string.

```
4. package main
5.
6. import "fmt"
7.
8. func main() {
9.     str := "viiikaas"
10.    count := 0
11.
12.    for _, char := range str {
13.        if char == 'a' || char == 'e' || char == 'i' || char == 'o' ||
14.        char == 'u' {
15.            count = count + 1
16.        }
17.    }
18.    fmt.Println("Number of vowels in ", str, " is", count)
19.}
```

Output:



The screenshot shows a PowerShell terminal window with the following content:

```
PS D:\go\sample\demo> go run vowels.go
Number of vowels in viiikaas is 5
PS D:\go\sample\demo>
```

The terminal window has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The output shows the program successfully counting 5 vowels in the string 'viiikaas'.

4. Go Program to print indices of the two numbers such that they add up to **target**.

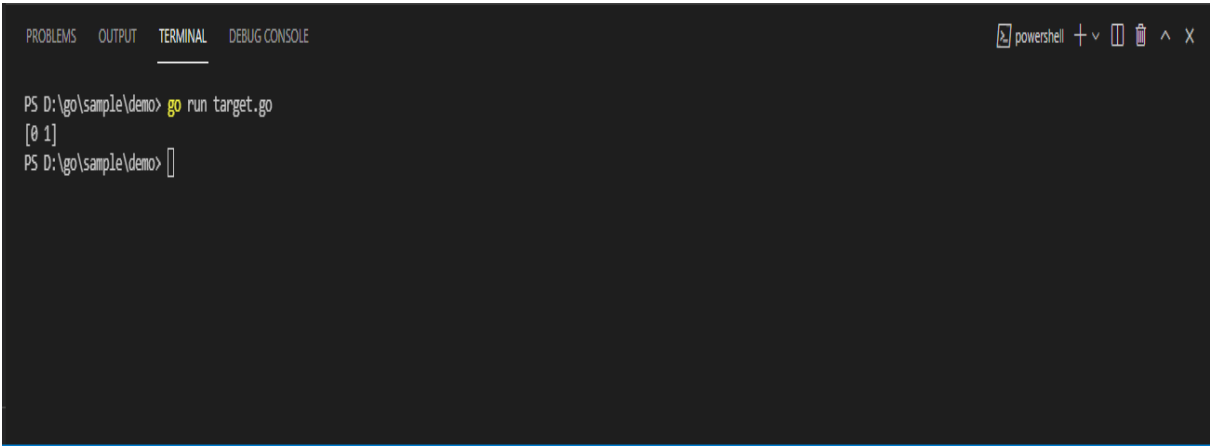
```
package main

import "fmt"

func main() {
    arr := []int{2, 7, 11, 15}
    twoSum(arr, 9)
}

func twoSum(nums []int, target int) {
    var a []int
    for i := 0; i < len(nums); i++ {
        for j := i + 1; j < len(nums); j++ {
            if nums[i]+nums[j] == target {
                a = append(a, i)
                a = append(a, j)
            }
        }
    }
    fmt.Println(a)
}
```

Output:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS D:\go\sample\demo> go run target.go
[0 1]
PS D:\go\sample\demo>
```

5. Go program to print median of two sorted arrays.

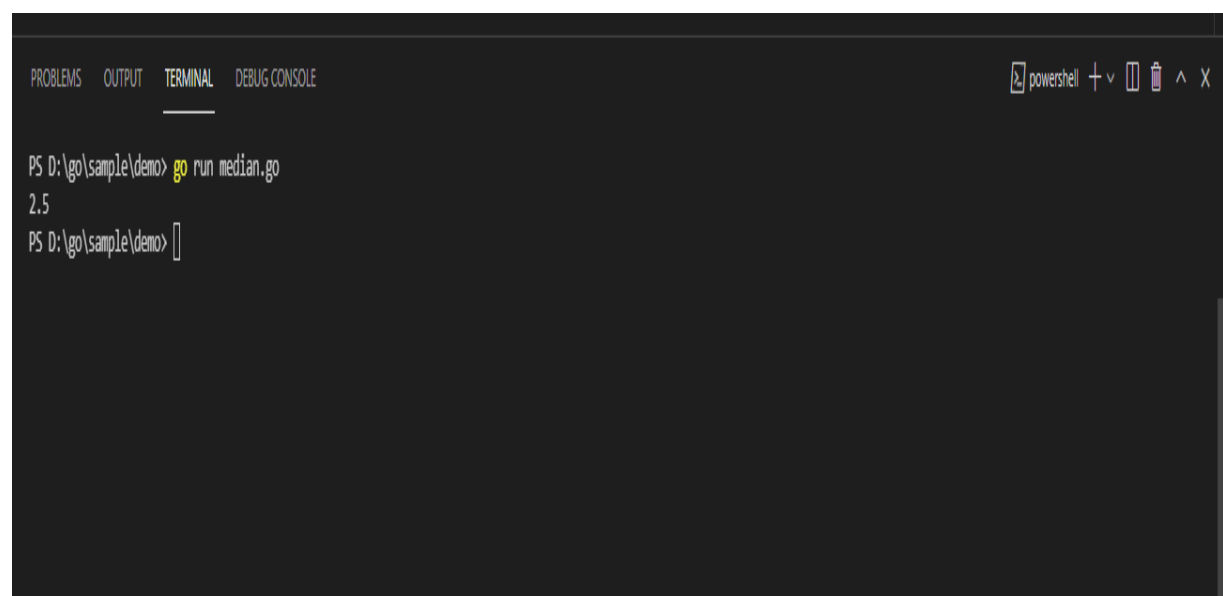
```
package main

import (
    "fmt"
    "sort"
)

func main() {
    arr1 := []int{1, 2}
    arr2 := []int{3, 4}
    findMedianSortedArrays(arr1, arr2)
}

func findMedianSortedArrays(nums1 []int, nums2 []int) {
    nums1 = append(nums1, nums2...)
    var ans float64
    n := len(nums1)
    sort.Ints(nums1)
    if n%2 != 0 {
        ans = float64(nums1[(n-1)/2])
    } else {
        ans = float64(nums1[(n-1)/2]+nums1[n/2]) / 2.0
    }
    fmt.Println(ans)
}
```

Output:



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS D:\go\sample\demo> go run median.go
2.5
PS D:\go\sample\demo>
```

6. Go program to reverse an integer.

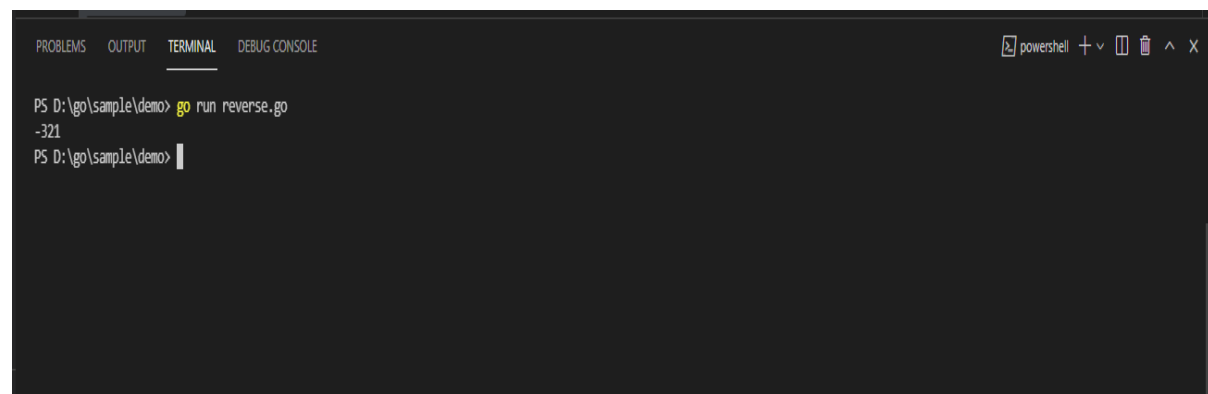
```
package main

import "fmt"

func main() {
    reverse(-123)
}

func reverse(x int) int {
    res := 0
    min := -2147483648
    max := 2147483647
    for x > 0 {
        num := x % 10
        res = res*10 + num
        x /= 10
    }
    for x < 0 {
        num := x % 10
        res = res*10 + num
        x /= 10
    }
    if res >= max || res <= min {
        return 0
    }
    fmt.Println(res)
    return res
}
```

Output:



The screenshot shows a PowerShell terminal window with the following content:

```
PS D:\go\sample\demo> go run reverse.go
-321
PS D:\go\sample\demo>
```

The terminal window has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The output shows the program successfully reversing the integer -123 to -321.

7. Go program to check a string contains a specified substring.

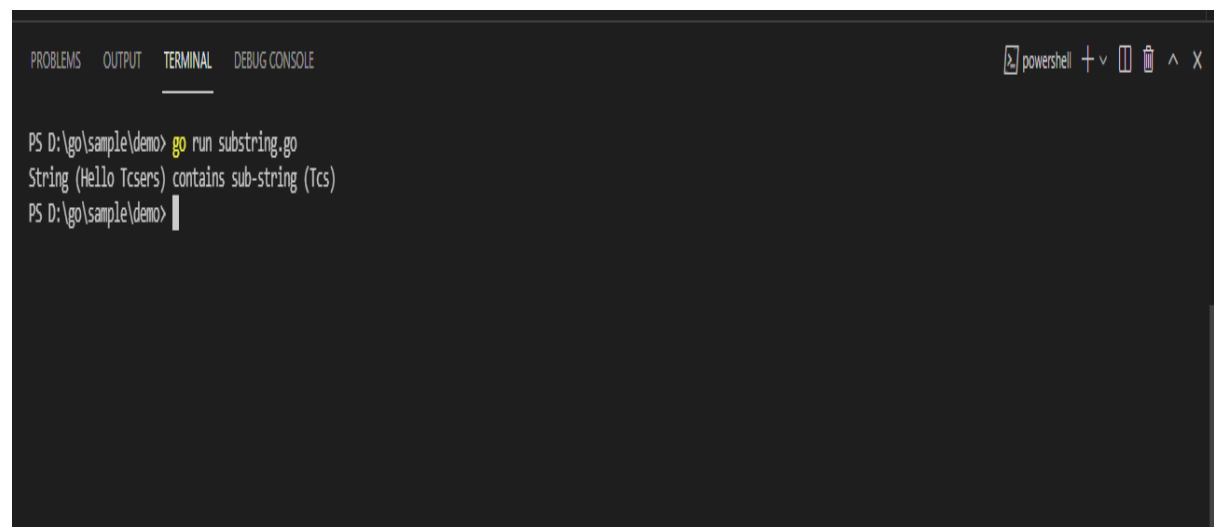
```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var str string = "Hello Tcsers"
    var subStr string = "Tcs"

    if strings.Contains(str, subStr) == true {
        fmt.Printf("String (%s) contains sub-string (%s)", str, subStr)
    } else {
        fmt.Printf("String (%s) does not contains substring (%s)", str, subStr)
    }
}
```

Output:



The screenshot shows a PowerShell terminal window with the following content:

```
PS D:\go\sample\demo> go run substring.go
String (Hello Tcsers) contains sub-string (Tcs)
PS D:\go\sample\demo>
```

The terminal window has tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The output shows the command `go run substring.go` being executed, followed by the output `String (Hello Tcsers) contains sub-string (Tcs)`.

8. Go program to print ascii values.

```
package main

import "fmt"

func main() {
    str := "vikas"

    fmt.Println("Ascii value of string:")
    for i := 0; i < len(str); i++ {
        fmt.Printf("%c %d\n", str[i], str[i])
    }
}
```

Output:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE powershell +v [ ] ^ X

PS D:\go\sample\demo> go run ascii.go
Ascii value of string:
v 118
i 105
k 107
a 97
s 115
PS D:\go\sample\demo> [ ]
```


9. Go program to sort an array using bubblesort.

```
package main

import "fmt"

func BubbleSort(array []int) []int {
    for i := 0; i < len(array)-1; i++ {
        for j := 0; j < len(array)-i-1; j++ {
            if array[j] > array[j+1] {
                array[j], array[j+1] = array[j+1], array[j]
            }
        }
    }
    return array
}

func main() {
    array := []int{11, 14, 3, 8, 18, 17, 43}
    fmt.Println(BubbleSort(array))
}
```

Output:

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS D:\go\sample\demo> go run bubblesort.go
[3 8 11 14 17 18 43]
PS D:\go\sample\demo>
```

10. Go program to search a element in an array using binary search.

```
package main

import "fmt"

func binarySearch(needle int, haystack []int) bool {

    low := 0
    high := len(haystack) - 1

    for low <= high {
        median := (low + high) / 2


        if haystack[median] < needle {
            low = median + 1
        } else {
            high = median - 1
        }
    }

    if low == len(haystack) || haystack[low] != needle {
        return false
    }

    return true
}

func main() {
    items := []int{1, 2, 9, 20, 31, 45, 63, 70, 100}
    fmt.Println(binarySearch(79, items))
}
```

Output:



```
PS D:\go\sample\demo> go run binary.go
true
PS D:\go\sample\demo> go run binary.go
false
PS D:\go\sample\demo>
```

