









**STRING TO INTEGER**

package main

import (

"fmt"

"strconv"

"strings"

"unicode"

)

func stripNonIntFloat(s string) string {

f := func(c rune) bool {

return !unicode.IsNumber(c) && (c != 46)

}

output := strings.FieldsFunc(s, f)

if len(output) > 0 {

return output[0]

} else {

return ""

}

}

func main() {

strList := []string{"10", "65.0", "xx", "11xx", "xx11"}

for i := 0; i < len(strList); i++ {

s := stripNonIntFloat(strList[i])

v, err := strconv.ParseFloat(s, 10)

if err != nil {

fmt.Println(strList[i], 0)

} else {

fmt.Println(strList[i], v)

}

}

}

**SUM OF 3 NUMBERS**

func threeSum(nums []int) [][]int {

var res [][]int

sort.Ints(nums)

lg := len(nums)

for i := 0; i < lg-2; i++ {

l := i + 1

r := lg - 1

if i > 0 && nums[i] == nums[i-1] {

continue

}

for l < r {

if nums[i]+nums[l]+nums[r] == 0 {

res = append(res, []int{nums[i], nums[l], nums[r]})

l++

r--

for l < r && nums[l] == nums[l-1] {

l++

}

for l < r && nums[r] == nums[r+1] {

r--

}

} else if nums[i]+nums[l]+nums[r] < 0 {

l++

} else {

r--

}

}

}

return res

}

**WATER TANK**

func maxArea(height []int) int {

capacity, l, r := 0, 0, len(height)-1

for l<r{

c := 0

if height[l] < height[r] {

c = height[l] \* (r-l)

l++

}else{

c = height[r] \* (r-l)

r--

}

if c > capacity {

capacity = c

}

}

return capacity

}

**SQUARE THE LIST**

unc sortedSquares(A []int) []int {

for i := range A {

A[i] = A[i]\*A[i]

}

qSort(A)

return A

}

func qSort(a []int) {

if len(a) < 2 {

return

}

p := len(a)/2

left, right := 0, len(a)-1

a[p], a[right] = a[right], a[p]

for i := range a {

if a[i] < a[right] {

a[left], a[i] = a[i], a[left]

left++

}

}

a[right], a[left] = a[left], a[right]

qSort(a[:left])

qSort(a[left+1:])

}