

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

var rome = map[string]int{
    "I":    1,
    "II":   2,
    "III":  3,
    "IV":   4,
    "V":    5,
    "VI":   6,
    "VII":  7,
    "VIII": 8,
    "IX":   9,
    "X":   10,
}

func parseValue(str string) (v_num int, b bool) {
    v_num, err := strconv.Atoi(str)
    if err != nil {
        v_num, ok := rome[str]
        if ok {
            return v_num, true
        } else {
            panic("Римское число не соответствует!")
        }
    }
    if v_num < 11 && v_num > 0 {
        return v_num, false
    } else {
        panic("Арабское число не входит в диапазон от 1 до 10")
    }
}

func arToRome(res int) string {
    var rome_2 = map[int]string{
        1:    "I",
        4:    "IV",
        5:    "V",
        9:    "IX",
        10:   "X",
        40:   "XL",
        50:   "L",
        90:   "XC",
    }

```

```

        100: "C",
    }
    keys := []int{100, 90, 50, 40, 10, 9, 5, 4, 1}
    roman := ""

    for _, key := range keys {
        for res >= key {
            roman += rome_2[key]
            res -= key
        }
    }
    return roman
}

func calculate(str string) string {
    arr := strings.Split(str, " ")
    v1, b := parseValue(arr[0])
    v2, b2 := parseValue(arr[2])
    if b != b2 {
        panic("Должны использоваться числа одного типа")
    }
    op := arr[1]
    var res int
    switch op {
    case "+":
        res = v1 + v2
    case "-":
        res = v1 - v2
    case "/":
        res = int(v1 / v2)
    case "*":
        res = v1 * v2
    }
    if b {
        if res <= 0 {
            panic("Результат операции над римскими числами не может быть меньше
или равен 0")
        } else {
            return arToRome(res)
        }
    }
    return strconv.Itoa(res)
}

func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Введите выражение: ")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)
    fmt.Println(calculate(input))
}

```

