**package main**

**import "fmt"**

**// Изменяем только данные слайса-параметра**

**func f1(c []int) {**

**for i, \_ := range c {**

**c[i] +=100**

**}**

**fmt.Println(c, len(c), cap(c))**

**}**

**func f2(pc \*[]int) {**

**for i, \_ := range \*pc {**

**(\*pc)[i] +=100**

**}**

**fmt.Println(\*pc, len(\*pc), cap(\*pc))**

**}**

**func f3(c []int) []int {**

**for i, \_ := range c {**

**c[i] +=100**

**}**

**fmt.Println(c, len(c), cap(c))**

**return c**

**}**

**func main() {**

**a:= [...]int {0,1,2,3,4,5,6,7,8,9}**

**fmt.Println(a) // [0 1 2 3 4 5 6 7 8 9]**

**arr:= a**

**b:= arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f1(b) // [102 103 104 105 106] 5 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106] 5 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f2(&b) // [102 103 104 105 106] 5 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106] 5 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**b = f3(b) // [102 103 104 105 106] 5 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106] 5 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 7 8 9]**

**}**

**01a.go**

**package main**

**import "fmt"**

**// Изменяем только длину слайса-параметра**

**func f1(c []int) {**

**c = c[:cap(c)]**

**fmt.Println(c, len(c), cap(c))**

**}**

**func f2(pc \*[]int) {**

**\*pc = (\*pc)[:cap(\*pc)]**

**fmt.Println(\*pc, len(\*pc), cap(\*pc))**

**}**

**func f3(c []int) []int {**

**c = c[:cap(c)]**

**fmt.Println(c, len(c), cap(c))**

**return c**

**}**

**func main() {**

**a:= [...]int {0,1,2,3,4,5,6,7,8,9}**

**fmt.Println(a) // [0 1 2 3 4 5 6 7 8 9]**

**arr:= a**

**b:= arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f1(b) // [2 3 4 5 6 7 8 9] 8 8**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f2(&b) // [2 3 4 5 6 7 8 9] 8 8**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 7 8 9] 8 8**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**b = f3(b) // [2 3 4 5 6 7 8 9] 8 8**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 7 8 9] 8 8**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**}**

**01b.go**

**package main**

**import "fmt"**

**// Изменяем длину слайса-параметра, а затем данные**

**func f1(c []int) {**

**c = c[:cap(c)]**

**for i, \_ := range c {**

**c[i] +=100**

**}**

**fmt.Println(c, len(c), cap(c))**

**}**

**func f2(pc \*[]int) {**

**\*pc = (\*pc)[:cap(\*pc)]**

**for i, \_ := range \*pc {**

**(\*pc)[i] +=100**

**}**

**fmt.Println(\*pc, len(\*pc), cap(\*pc))**

**}**

**func f3(c []int) []int {**

**c = c[:cap(c)]**

**for i, \_ := range c {**

**c[i] +=100**

**}**

**fmt.Println(c, len(c), cap(c))**

**return c**

**}**

**func main() {**

**a:= [...]int {0,1,2,3,4,5,6,7,8,9}**

**fmt.Println(a) // [0 1 2 3 4 5 6 7 8 9]**

**arr:= a**

**b:= arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f1(b) // [102 103 104 105 106 107 108 109] 8 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106] 5 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 107 108 109]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f2(&b) // [102 103 104 105 106 107 108 109] 8 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106 107 108 109] 8 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 107 108 109]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**b = f3(b) // [102 103 104 105 106 107 108 109] 8 8**

**fmt.Println(b, len(b), cap(b)) // [102 103 104 105 106 107 108 109] 8 8**

**fmt.Println(arr) // [0 1 102 103 104 105 106 107 108 109]**

**}**

**01c.go**

**package main**

**import "fmt"**

**// Добавляем в слайс-параметр данные по одному,**

**// пока не произойдёт перераспределения памяти**

**func f1(c []int) {**

**ccap:= cap(c)**

**for cap(c) == ccap {**

**c = append(c, len(c)\*10)**

**}**

**fmt.Println(c, len(c), cap(c))**

**}**

**func f2(pc \*[]int) {**

**ccap:= cap(\*pc)**

**for cap(\*pc) == ccap {**

**\*pc = append(\*pc, len(\*pc)\*10)**

**}**

**fmt.Println(\*pc, len(\*pc), cap(\*pc))**

**}**

**func f3(c []int) []int {**

**ccap:= cap(c)**

**for cap(c) == ccap {**

**c = append(c, len(c)\*10)**

**}**

**fmt.Println(c, len(c), cap(c))**

**return c**

**}**

**func main() {**

**a:= [...]int {0,1,2,3,4,5,6,7,8,9}**

**fmt.Println(a) // [0 1 2 3 4 5 6 7 8 9]**

**arr:= a**

**b:= arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f1(b) // [2 3 4 5 6 50 60 70 80] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**fmt.Println(arr) // [0 1 2 3 4 5 6 50 60 70]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f2(&b) // [2 3 4 5 6 50 60 70 80] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 50 60 70 80] 9 16**

**fmt.Println(arr) // [0 1 2 3 4 5 6 50 60 70]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**b = f3(b) // [2 3 4 5 6 50 60 70 80] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 50 60 70 80] 9 16**

**fmt.Println(arr) // [0 1 2 3 4 5 6 50 60 70]**

**}**

**01d.go**

**package main**

**import "fmt"**

**// Добавляем в слайс-параметр данные группой, размер**

**// которой требует перераспределения памяти**

**func f1(c []int) {**

**cc:= make([]int, cap(c) - len(c) + 1)**

**c = append(c, cc...)**

**fmt.Println(c, len(c), cap(c))**

**}**

**func f2(pc \*[]int) {**

**cc:= make([]int, cap(\*pc) - len(\*pc) + 1)**

**\*pc = append(\*pc, cc...)**

**fmt.Println(\*pc, len(\*pc), cap(\*pc))**

**}**

**func f3(c []int) []int {**

**cc:= make([]int, cap(c) - len(c) + 1)**

**c = append(c, cc...)**

**fmt.Println(c, len(c), cap(c))**

**return c**

**}**

**func main() {**

**a:= [...]int {0,1,2,3,4,5,6,7,8,9}**

**fmt.Println(a) // [0 1 2 3 4 5 6 7 8 9]**

**arr:= a**

**b:= arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f1(b) // [2 3 4 5 6 0 0 0 0] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**f2(&b) // [2 3 4 5 6 0 0 0 0] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 0 0 0 0] 9 16**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**arr = a**

**b = arr[2:7]**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6] 5 8**

**b = f3(b) // [2 3 4 5 6 0 0 0 0] 9 16**

**fmt.Println(b, len(b), cap(b)) // [2 3 4 5 6 0 0 0 0] 9 16**

**fmt.Println(arr) // [0 1 2 3 4 5 6 7 8 9]**

**}**

**01e.go**

**package main**

**// Запуск из командной строки:**

**// go test -bench . benchmark\_sample\_test.go**

**// В данном случае benchmark\_sample\_test.go - это имя файла,**

**// в котором находится данная программа. Имя тестируемого**

**// файла обязательно должно заканиваться на \_test**

**import "testing"**

**func lala(s []int) {**

**for i, x := range(s) {**

**s[i] = (x-1)\*x\*(x+1)**

**}**

**}**

**// Названия тестируемых функций должны начинаться на Benchmark,**

**// за которым идёт название, начинающееся с большой буквы**

**// Оценивается функция BenchmarkLala.**

**func BenchmarkLala(b \*testing.B) {**

**var s [100000]int**

**for i:= 0; i < b.N; i++ {**

**lala(s[:])**

**}**

**}**

**// Оценивается функция BenchmarkLalala.**

**func BenchmarkLalala(b \*testing.B) {**

**var s [100000]int**

**for k:= 0; k < b.N; k++ {**

**for i, x := range s {**

**s[i] = (x-1)\*x\*(x+1)**

**}**

**}**

**}**

**// Оценивается функция BenchmarkLalala2.**

**func BenchmarkLalala2(b \*testing.B) {**

**s:= make([]int, 100000)**

**for k:= 0; k < b.N; k++ {**

**for i, x := range s {**

**s[i] = (x-1)\*x\*(x+1)**

**}**

**}**

**}**

**// OUTPUT**

**// goos: windows**

**// goarch: amd64**

**// BenchmarkLala-4 12103 99203 ns/op**

**// BenchmarkLalala-4 7063 161336 ns/op**

**// BenchmarkLalala2-4 12094 98955 ns/op**

**// PASS**

**// ok command-line-arguments 5.765s**

**benchmark\_sample\_test.go**

**package main**

**import (**

**"testing"**

**"fmt"**

**)**

**func lala(s []int) {**

**for i, x := range(s) {**

**s[i] = (x-1)\*x\*(x+1)**

**}**

**}**

**func main() {**

**var s [100000]int**

**res:= testing.Benchmark(**

**func(b \*testing.B) {**

**for i:= 0; i < b.N; i++ {**

**lala(s[:])**

**}**

**})**

**fmt.Println(res)**

**}**

**benchmark\_sample\_02.go**

**// Запуск:**

**// go test -bench . insert\_test.go**

**// файл должен иметь суффикс \_test**

**// оцениваемые функции должны начинаться с Benchmark,**

**// за которым идёт название, начинающееся с большой буквы**

**package insert**

**import (**

**"container/list"**

**"testing"**

**)**

**var (**

**slice10 = slice(10)**

**slice100 = slice(100)**

**slice1k = slice(1000)**

**slice10k = slice(10000)**

**slice100k = slice(100000)**

**slice1m = slice(1000000)**

**)**

**func slice(size int) []byte { return make([]byte, size) }**

**func BenchmarkInsertHeadSlice10(b \*testing.B) {**

**benchmarkInsertSlice(b, slice10) }**

**func BenchmarkInsertHeadSlice100(b \*testing.B) {**

**benchmarkInsertSlice(b, slice100) }**

**func BenchmarkInsertHeadSlice1000(b \*testing.B) {**

**benchmarkInsertSlice(b, slice1k) }**

**func BenchmarkInsertHeadSlice10000(b \*testing.B) {**

**benchmarkInsertSlice(b, slice10k) }**

**func BenchmarkInsertHeadSlice100000(b \*testing.B) {**

**benchmarkInsertSlice(b, slice100k) }**

**func BenchmarkInsertHeadSlice1000000(b \*testing.B) {**

**benchmarkInsertSlice(b, slice1m) }**

**func benchmarkInsertSlice(b \*testing.B, a []byte) {**

**c := cap(a)**

**for n := 0; n < b.N; n++ {**

**// insert at i trick:**

**// a = append(a[:i], append([]T{x}, a[i:]...)...)**

**// could be simplified as we insert to head**

**a = append([]byte{0}, a...)**

**// reset a**

**a = a[:c:c]**

**}**

**}**

**var (**

**linkedList10 = linkedList(10)**

**linkedList100 = linkedList(100)**

**linkedList1k = linkedList(1000)**

**linkedList10k = linkedList(10000)**

**linkedList100k = linkedList(100000)**

**linkedList1m = linkedList(1000000)**

**)**

**func linkedList(size int) \*list.List {**

**l := list.New()**

**for i := 0; i < size; i++ {**

**l.PushBack(0)**

**}**

**return l**

**}**

**func BenchmarkInsertHeadList10(b \*testing.B) {**

**benchmarkInsertList(b, linkedList10) }**

**func BenchmarkInsertHeadList100(b \*testing.B) {**

**benchmarkInsertList(b, linkedList100) }**

**func BenchmarkInsertHeadList1000(b \*testing.B) {**

**benchmarkInsertList(b, linkedList1k) }**

**func BenchmarkInsertHeadList10000(b \*testing.B) {**

**benchmarkInsertList(b, linkedList10k) }**

**func BenchmarkInsertHeadList100000(b \*testing.B) {**

**benchmarkInsertList(b, linkedList100k) }**

**func BenchmarkInsertHeadList1000000(b \*testing.B) {**

**benchmarkInsertList(b, linkedList1m) }**

**func benchmarkInsertList(b \*testing.B, l \*list.List) {**

**for n := 0; n < b.N; n++ {**

**l.PushFront(0)**

**// reset l**

**l.Remove(l.Front())**

**}**

**}**

**insert\_test.go**