

# Programmazione

(Vers. B)

## Appello di Gennaio 2019

Cognome ..... Nome .....  
Matricola ..... Anno di corso ..... DSA

Alcune delle seguenti domande riguardano il vostro numero di matricola. Scrivete il vostro numero nel seguente schema, una cifra per ogni spazio:

A	B	C	D	E	F

Nel seguito, useremo le lettere ABC... per riferirci alle corrispondenti cifre del numero di matricola. Ad esempio, "il numero DEF" il numero costituito dalle ultime tre cifre del vostro numero di matricola.

Rispondete alle seguenti domande:

1. Assumete che  $x$  e  $y$  siano variabili **int**, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente AF e BE. Assumete inoltre che  $z$  sia una variabile di tipo **int** e di valore  $\{A,B,C,D,E,F\}$ . Dite quale sarà il valore di  $x$  e  $y$  *dopo* gli assegnamenti indicati<sup>1</sup>

Assegnamento	x	y
$x += y$		
$y = x + z[1]$		
$x *= z[y\%6]$		
$x, _ = \text{strconv.Atoi}("5" + \text{strconv.Itoa}(y))$		
$x, y = (10 * z[0] + z[3]) / (y / 10), y - 10$		

<sup>1</sup>Il pacchetto `strconv` contiene la funzione `Itoa(x int) string` che converte l'intero  $x$  in una stringa (la sua rappresentazione in base 10). La funzione `Atoi(s string) (int, error)` effettua la conversione inversa, restituendo eventualmente un errore.

2. Scrivete una funzione  $f$  che data una slice di interi, restituisca il valore ottenuto sommandone gli elementi di indice dispari.

3. Considerate la seguente funzione:

```
func f(x []int, y int) int {
    a := 0
    b := y - 1
    for _, c := range x {
        if c >= b {
            a++
        }
    }
    return a
}
```

I nomi delle tre variabili sono scelti molto male. Dopo aver capito che cosa fa la funzione, scegliete dei nomi migliori per le variabili  $a$ ,  $b$  e  $c$ , associando a ciascuna delle variabili che compaiono nella tabella di sinistra uno dei nomi della tabella di destra

a
b
c

yMenoUno
indice
flag
successivo
contatore
i
elemento

Inoltre, dite che cosa restituisce la funzione dandole come primo argomento la slice `[]int{A,B,C,D,E}` e come secondo argomento i seguenti valori (che dipendono dalle cifre del vostro numero di matricola):

<b>Argomento</b>	<b>Valore restituito</b>
A	
B	
C	
D	
E	

4. Volete scrivere una funzione che, data una slice di stringhe ASCII, determina in quante di esse compare una certa runa: non importa quante volte compare nella singola stringa, ma solo in quante stringhe compare.. Ad esempio, se la slice contiene le sole stringhe {"cane", "cene", "katai", "telone", "rosi"} e la runa è 'a', il valore della funzione sarà 2.

Riempite le parti mancanti:

```
func f(s ....., c rune) ..... {
    count := 0
    for _, str := range s {
        var i int
        n := len(.....)
        for i = 0; i < n; i++ {
            if rune(str[i]) == c {
                break
            }
        }
        if ..... {
            count++
        }
    }
    return count
}
```

5. Considerate la seguente funzione ricorsiva:

```

func f(a int, b int) int {
    if b == 0 {
        return a
    }
    return f(10 * a + b % 10, b / 10)
}

```

- Che valore restituisce la chiamata f(ABCA, AB)? .....
- Che valore restituisce la chiamata f(AB, ABACA)? .....

6. Qualcuno ha scritto la seguente funzione, ma purtroppo non è molto leggibile. Supponete di volerla chiamare passando come argomenti i valori EA e ACB (dal vostro numero di matricola).

```

func f(x int, y int) (int, int) {
    m := x < y
    if m {
        x, y = y, x
    }
    t := 0
    for x > y {
        x -= y
        t++
    }
    return x, t
}

```

Considerate le variabili elencate nella tabella e, per ciascuna, dite quale è il suo tipo e il suo valore alla fine della funzione, prima che venga eseguito il **return**.

Variabile	Tipo	Valore
x		
y		
m		
t		

7. Nel seguente frammento di codice Go, viene definito un tipo `persona` e una funzione che data una slice di persone cerca quante sono quelle il cui cognome inizia con un certo prefisso. Per esempio, se il prefisso è *Ros* conterà quante sono le persone il cui cognome inizia per *Ros* (per esempio, *Rossi*, *Rosati*, *Rosi* ecc.).

Purtroppo la funzione contiene 3 errori. Trovateli, cerchiateli, e ricopiate nella tabella sotto l'espressione (o l'istruzione) errata e quella corretta.

```
type persona {
    cognome, nome string
}

func findPref(a []persona, prefisso string) int {
    c := 0
    n := len(a)
    for _, x := range a {
        if x.cognome[:n] = prefisso {
            c++
        }
    }
    return c
}
```

1		
2		
3		

8. Scrivete una funzione che, presi come argomenti due interi  $x$  e  $y$ , restituisca una slice di interi che contenga i divisori di  $x$  che non sono divisori di  $y$ .

9. Considerate le seguenti funzioni.

```

func f(z []int) bool {
    c := true
    for _, v := range z {
        if v % 2 == 0 {
            c = !c
        }
    }
    return c
}

func g(x int) bool {
    return f([]int{2 * x + 1, x})
}

func h(x int, y int) bool {
    return g(x + 1) && g(y)
}

```

Siano ora  $X$  e  $Y$  due numeri interi positivi, e  $Z$  una slice di numeri interi. Per ciascuna delle seguenti affermazioni, dite se è vera o falsa (indicando V o F nell'apposita casella).

Affermazione	V/F
$h(3, 5)$ restituisce <b>false</b>	
$g(X)$ restituisce <b>true</b> se e solo se $X$ è dispari	
$g(X)$ restituisce <b>true</b> se e solo se $X$ è pari	
c'è almeno un valore di $X$ per cui $h(X, X)$ restituisce <b>false</b> e almeno un valore di $X$ per cui restituisce <b>true</b>	
c'è almeno un valore di $Z$ per cui $f(Z)$ restituisce <b>false</b> e almeno un valore di $Z$ per cui restituisce <b>true</b>	
se $W$ è ottenuta concatenando $Z$ a se stessa, $f(W)$ restituisce <b>true</b>	