

# Programmazione

(Vers. A)

## Appello di Giugno 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  $\{AB, BA, CB, DB, EA, FF\}$ . Dite quale sarà il valore di  $x$  e  $y$  *dopo* gli assegnamenti indicati<sup>1</sup>

Assegnamento	x	y
$x, y = z[(y+1)\%6], z[(y+2)\%6]$		
$y, - = \text{strconv.Atoi}(\text{strconv.Itoa}(z[y\%2]) + \text{strconv.Itoa}(z[(y/2)\%6]))$		
$x = z[2] - z[(1+y*5)\%6]$		
$x, - = \text{strconv.Atoi}("1" + \text{strconv.Itoa}(z[0] + 1))$		
$z[0], y = y + z[0], x = z[0]$		

<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 stringhe, restituisca una slice di interi della stessa lunghezza che contenga nella  $i$ -esima posizione il numero di 'a' (a minuscole) della  $i$ -esima stringa. Ad esempio, invocandola sulla slice `[]string{"cane", "tana", "", "dormitorio"}` dovrebbe restituire `[]int{1,2,0,0}`.

3. Considerate la seguente funzione:

```

func f(x []int, y []string) int {
    c := 0
    for _, a := range y {
        va := int(a[0] - '0')
        for _, b := range x {
            for b > 0 {
                if b%10 == va {
                    c += va
                }
                b /= 10
            }
        }
    }
    return c
}

```

Dite che cosa restituisce la funzione dandole come primo argomento la slice `[]int{AB,BC,CD,DE,EF,FA}` e come secondo argomento i seguenti valori (che dipendono dalle cifre del vostro numero di matricola):

Argomento	Valore restituito
<code>[]string{"0", "1", "0", "1"}</code>	
<code>[]string{"2", "4", "6", "8", "2", "2"}</code>	
<code>[]string{"10", "20", "1", "2", "3", "4", "5"}</code>	
<code>[]string{"9", "8", "7", "90", "80"}</code>	
<code>[]string{}</code>	

4. Volete scrivere una funzione che dato un intero (che può essere positivo o negativo, ma non è mai nullo) restituisce il numero di caratteri da cui è costituita la sua rappresentazione decimale (tale lunghezza è il numero di cifre per i numeri positivi, e il numero di cifre più uno per i negativi).

Riempite le parti mancanti:

```
func f(s ..... ) int {
    count := 0
    if ..... {
        count++
        s *= -1
    }
    for ..... {
        s /= 10
        .....
    }
    return count
}
```

5. Considerate la seguente funzione ricorsiva:

```
func f(a int) int {
    if a == 0 {
        return 0
    } else {
        return f(a / 10) + a % 10;
    }
}
```

- Che valore restituisce la chiamata f(FACAA)? .....
- Che valore restituisce la chiamata f(CEF)? .....

6. Qualcuno ha scritto la seguente funzione, ma purtroppo non è molto leggibile. Supponete di volerla chiamare passando come argomenti il valore 2A (A è preso dal vostro numero di matricola).

```
func f(x int) int {
    var r, c, d int
    for c = 2; c <= x; c++ {
        for d = 2; d < c; d++ {
            if c % d == 0 {
                break
            }
        }
        if d == c {
            r++
        }
    }
    return r
}
```

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
r		
c		
d		
x		

7. Nel seguente frammento di codice Go, viene definito un tipo esperimento che è una struttura che contiene vari campi fra cui una slice di interi (chiamati dati). E' inoltre definita una funzione contaNeg che data una slice di esperimenti, conta quanti di essi hanno la caratteristica di contenere solo dati negativi.

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

```
type esperimento struct {
    nome string
    dati []int
}

func contaNeg(x []esperimento) int {
    c := 0
    for _, e := range x {
        c++
        for _, d := range dati {
            if d < 0 {
                c--
                continue
            }
        }
    }
    return c
}
```

1		
2		
3		

8. Scrivete una funzione  $f$  che, presi come argomenti tre interi  $x, y, z$ , restituisca una slice di interi che contenga (in quest'ordine)  $x, x + y, x + 2y$  eccetera fino a  $z$  (escluso). Ad esempio,  $f(5, 3, 17)$  dovrebbe essere la slice

`[] int{5, 8, 11, 14}`

9. Considerate le seguenti funzioni.

```

func f(z [] int) (int, int, int) {
    var a [3] int
    for _, v := range z {
        if v < 0 {
            a[0]++
        } else if v == 0 {
            a[1]++
        } else {
            a[2]++
        }
    }
    return a[0], a[1], a[2]
}

func g(x int) (int, int) {
    a, _, c := f([] int{x, -x})
    return a, c
}

func h(x int) int {
    _, b, _ := f([] int{x, x + 1, x + 2})
    return b
}

```

Siano ora  $X$  un numero intero 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
la funzione $f(Z)$ può restituire valori negativi	
la somma dei valori restituiti da $f(Z)$ è uguale alla lunghezza di $Z$	
i due valori restituiti da $g(X)$ possono essere diversi fra loro	
esiste un $X$ per cui $g(X)$ restituisce 0, 0	
ci sono esattamente tre valori di $X$ per cui $h(X)$ restituisce 1	
la funzione $h(X)$ può restituire valori maggiori di 1	