

Programmazione

(Vers. A)

Appello di Febbraio 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 indicati¹

Assegnamento	x	y
$x += z[(y+5)\%6]$		
$y, _ = \text{strconv.Atoi}(\text{strconv.Itoa}(y) + \text{strconv.Itoa}(y))$		
$x *= z[y \% 2] - z[y \% 3]$		
$x, _ = \text{strconv.Atoi}("1" + \text{strconv.Itoa}(1 + y))$		
$x, y = y + z[1], x - z[0]$		

¹Il 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 sommando le cifre delle unità più uno di tutti gli elementi della slice. Ad esempio, invocandola sulla slice `[]int{15, 6, 30, 21}` dovrebbe restituire $(5 + 1) + (6 + 1) + (0 + 1) + (1 + 1) = 6 + 7 + 1 + 2 = 16$.

3. Considerate la seguente funzione:

```
func f(x []int, y []string) int {
    c := 0
    for _, a := range x {
        for _, b := range y {
            if strconv.Itoa(a) == b {
                c += a
                break
            }
        }
    }
    return c
}
```

Dite che cosa restituisce la funzione dandole come primo argomento la slice `[]int{A,B,C,D,E,F}` 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{"ciao", "9", "8", "7", "90", "80"}</code>	
<code>[]string{}</code>	

4. Volete scrivere una funzione che, data una slice di stringhe ASCII, determina quante di esse sono costituite solo da una certa runa, eventualmente ripetuta molte volte. Ad esempio, se la slice contiene le sole stringhe {"cane", "aaa", "bbb", "taana", "a"} e la runa è 'a', il valore della funzione sarà 2 (ci sono solo due stringhe nella slice che siano costituite dalla sola runa a).

Riempite le parti mancanti:

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

5. Considerate la seguente funzione ricorsiva:

```

func f(a int, b int) int {
    if b == 0 {
        return a
    }
    return 1 + f(a, b / 10)
}

```

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

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) float64 {
    var res float64
    c := 0
    for y > 0 {
        res = float64(y % 10) / 10.0 + res / 10.0
        y /= 10
        c++
    }
    res += float64(x % c)
    return res
}

```

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
res		
c		
x		
y		

7. Nel seguente frammento di codice Go, viene definito un tipo `data` e una funzione che data una slice di date cerca quante sono quelle estive. L'estate inizia il 21 giugno (compreso) e termina il 22 settembre (compreso).

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

```
type data struct {
    g, m, a int
}

func countSummer(x []data) int {
    c := 0
    for _, d := range x {
        switch x.m {
            case 7 || 8:
                c++
            case 6:
                if d.g >= 21 {
                    c++
                }
            case 9:
                if d.g >= 22 {
                    c++
                }
        }
    }
    return c
}
```

1		
2		
3		

8. Scrivete una funzione f che, presi come argomenti due interi x e y , restituisca una slice di stringhe che contiene tutte le stringhe costituite da al massimo x asterischi (anche zero va bene) seguiti da al massimo y punti esclamativi (anche zero va bene). Non è importante l'ordine in cui tali stringhe compaiono nella slice, basta che vi compaiano tutte e senza ripetizioni. Ad esempio, $f(5, 3)$ dovrebbe essere la slice

```
{ "", "!", "!!", "!!!", "*", "*!", "*!!", "*!!!", "**", "**!", "**!!", "**!!!",
"***", "***!", "***!!", "***!!!", "****", "****!", "****!!", "****!!!",
"*****", "*****!", "*****!!", "*****!!!" }
```

9. TODO Considerate le seguenti funzioni.

```

func f(z []int) int {
    c := 1
    for _, v := range z {
        if v < 0 {
            c *= -1
        }
        if v == 0 {
            c *= 0
        }
    }
    return c
}

func g(x int) int {
    return f([]int{x, -x})
}

func h(x int) int {
    return g(x + 1) + g(x)
}

```

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
$g(X)$ può valere 1	
$h(X)$ può valere 0	
$g(X)$ può valere 0	
$g(X)$ può valere -1	
$f(Z)$ vale sempre 0, 1 o -1	
$h(X)$ può valere -2	