

Programmazione

(Vers. D)

Appello di Febbraio 2020

Cognome Nome

Matricola Anno di corso (1, 2, ...) 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:

- Assumete che x e y siano variabili **int**, e che il loro valore *prima di ciascuno dei seguenti assegnamenti* sia rispettivamente DF e EF. Assumete inoltre che z sia una variabile di tipo `[]int` e di valore `[]int{A,B,C,D,E,F}`. Dite quale sarà il valore di x e y *dopo* gli assegnamenti indicati.¹

Assegnamento	x	y
<code>x += z[y % 6] - z[1] + y / 3</code>		
<code>x, y = y + z[y % 6], z[3] * x</code>		
<code>x, _ = strconv.Atoi(strconv.Itoa(z[(y + 4) % 6]) + "3")</code>		
<code>y, x = x + y, x - y</code>		
<code>x, y = y / 3, z[x % 3] % 3</code>		

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 che data una runa c , un intero x e una slice di stringhe s restituisca il numero di elementi della slice che contengono un numero di c maggiore o uguale di x . Per esempio, se la runa è 'a', il numero è 2 e la slice è {"ciao", "zona", "mamma_guarda", "pippo", "lana"} la funzione deve restituire 2 (perché *mamma guarda* e *lana* contengono entrambi due o più 'a').

3. Considerate la seguente funzione:

```
func f(x []int, y int) int {  
    a := 0  
    for i := 0; i < len(x) - 1; i++ {  
        if x[i] < 20 * y && x[i + 1] % (y + 1) != 0 {  
            a++  
        }  
    }  
    return a  
}
```

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:

Secondo argomento	Valore restituito
A	
B	
C	
D	
E	

4. Volete scrivere una funzione che, data una mappa da stringhe a stringhe, restituisce la concatenazione delle chiavi il cui valore associato coincida con la chiave. Non importa l'ordine di concatenazione. Per esempio, su input

```
{"ciao": "ciao", "ciao_mamma": "ciao", "mamma_guarda": "mamma_guarda", "pippo": "puppo"}
```

la funzione deve restituire "ciaomamma_guarda" oppure "mamma_guardaciao".

Riempite le parti mancanti:

```
func h(m ..... ) ..... {  
    conc := .....  
    for s, v := ..... {  
        if ..... {  
            conc += .....  
        }  
    }  
    return conc  
}
```

5. Considerate la seguente funzione ricorsiva:

```
func r(a int) int {
    if a < 0 {
        return -1
    }
    return r(a - 2) + 1
}
```

- Che valore restituisce la chiamata $f(DE)$?
- Che valore restituisce la chiamata $f(D)$?

6. Considerate la seguente funzione, e supponete di volerla chiamare passando come argomenti i valori ED, EF e B (dal vostro numero di matricola).

```
func t(x int, y int, z int) (int, bool) {
    m := (x % 3 == y / 3)
    t := m && (z % 2 != 0)
    if !t {
        x, y = y + 3, x
    } else {
        x, y = x + 1, y
    }
    x -= y
    return x, t
}
```

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

Variabile	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 calcola la frazione di donne con stipendio che supera una certa soglia (rispetto a tutte le persone il cui stipendio supera quella soglia). Ad esempio, se ci sono 52 persone con stipendio sotto la soglia, e di queste 10 sono donne, restituirà circa 0.1923 (che è 10 diviso 52).

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

```
type persona struct {
    cognome, nome string
    stipendio int
    m bool // m=true sse e' un uomo
}

func perc(a []persona, soglia int) float64 {
    c := 0
    cm := 0
    for p := range a {
        c++
        if p.stipendio > soglia {
            if p.m {
                cm++
            }
        }
    }
    return float64((c - cm) / c)
}
```

1		
2		
3		

8. Scrivete una funzione che presi come argomenti due slice di stringhe x e y restituisca una slice contenente le stringhe di x che siano prefisso di almeno una stringa di y . (Una stringa a è un prefisso della stringa b se i primi $\mathbf{len}(a)$ byte di b coincidono con i corrispondenti byte di a)

9. Considerate le seguenti funzioni.

```
func a(s []int) int {
    c := 0
    for _, x := range s {
        if x <= 1 {
            c++
        }
    }
    return c
}
```

```
func b(s []int) int {
    return a(append(s, 0))
}
```

```
func c(s []int, t []int) int {
    return b(s) + b(append(t, 1))
}
```

Sia ora X una slice di interi. Per ciascuna delle seguenti affermazioni, dite se è vera o falsa (indicando V o F nell'apposita casella). Notate che con “maggiore” e “minore” si intende la versione stretta (“strettamente maggiore”, “strettamente minore”).

c'è almeno un valore di X per cui $c(X, X)$ restituisce 3	
Affermazione	V/F
$c([\text{int}\{3,1\}, [\text{int}\{0,2\}])$ restituisce 5	
c'è almeno un valore di X per cui $c(X, X)$ restituisce 6	
$b(X)$ restituisce un valore maggiore di due se e solo se X contiene almeno due interi minori di due	
$b(X)$ restituisce un valore pari se e solo se X non contiene uni	
se W è ottenuta concatenando X a se stessa, $b(W)$ restituisce un numero dispari	
c'è almeno un valore di X per cui $b(X)$ restituisce zero e almeno un valore di X per cui restituisce un valore maggiore di zero	