

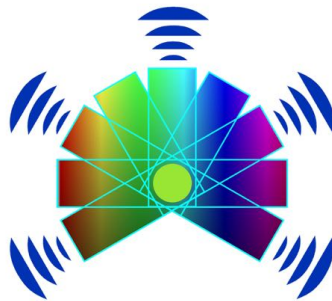


UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

Tutorato di programmazione Azzeramento

Ultimo aggiornamento 4 maggio 2020 - 14:04:45

Piano Lauree Scientifiche - Progetto di Informatica



Il materiale didattico per il tutorato di programmazione è stato preparato da un gruppo di lavoro formato da docenti e studenti tutor del Dipartimento di Informatica dell'Università degli Studi di Milano.

Docenti:

Violetta Lonati (coordinatrice del gruppo) e Anna Morpurgo

Studenti tutor:

Alexandru David, Andrea Zubenko, Davide Cernuto, Leonardo Albani, Margherita Pindaro, Umberto Costantini, Vittorio Peccenati.

Si ringraziano per i numerosi spunti:

Carlo Bellettini, Paolo Boldi, Mattia Monga, Massimo Santini e Sebastiano Vigna.

1 Conversione di una temperatura

Considerate il seguente programma `gradi.go`. Il programma legge una temperatura in gradi Fahrenheit e stampa l'equivalente in gradi centigradi.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var fahr float64
7     const FATTORE_SCALA, ZERO = 5.0 / 9.0, 32.0
8
9     fmt.Println("Inserisci la temperatura in gradi fahrenheit: ")
10    fmt.Scan(&fahr)
11    cels := (fahr - ZERO) * FATTORE_SCALA
12    fmt.Println("Gradi Celsius equivalenti:", cels)
13 }
```

Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande. Se avete dubbi, potete testarlo, eseguendolo su casi di input significativi e modificandolo.

1. A cosa serve l'istruzione alla riga 3?
2. A cosa serve l'istruzione alla riga 6?
3. Perché `FATTORE_SCALA` e `ZERO` sono dichiarate `const`?
4. Quanto valgono `FATTORE_SCALA`, `ZERO` e `fahr` quando viene eseguita l'istruzione alla riga 9?
5. A cosa serve l'istruzione alla riga 9?
6. A cosa serve l'istruzione alla riga 10?
7. Nella definizione di riga 7, è possibile sostituire `5.0` con `5`? È possibile sostituire `9.0` con `9`? È possibile sostituirli entrambi contemporaneamente? Giustificate la risposta.
8. Cosa fa l'istruzione alla riga 11?
9. Di che tipo è la variabile `cels`? Giustificate la risposta.

2 Tre casi

Considerate la seguente porzione del programma `tre_casi.go`.

```
1 func main() {
2     var n int
3
4     if len(os.Args) != 4 {
5         fmt.Println("lanciare il programma con tre argomenti")
6         return
7     }
8
9     fmt.Println("Inserisci un numero:")
10    _, err := fmt.Scan(&n)
11    if err != nil {
12        fmt.Println("ci vuole un numero intero")
13        return
14    }
15
16    switch n {
17    case 1:
18        os.Args[2], os.Args[3] = os.Args[3], os.Args[2]
19    case 2:
20        os.Args[1], os.Args[3] = os.Args[3], os.Args[1]
21    case 3:
22        os.Args[1], os.Args[2] = os.Args[2], os.Args[1]
23    }
24    fmt.Println(os.Args[1], os.Args[2], os.Args[3])
25 }
```

Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande. Se avete dubbi, potete testarlo, eseguendolo su casi di input significativi e modificandolo.

1. Cosa si aspetta in input il programma? Distinguate tra standard input e argomenti da linea di comando.
2. Come viene chiamato il simbolo '_' in Go? A cosa serve?
3. A cosa serve la variabile `err` nella riga 10?
4. Cosa fa l'istruzione alla riga 13?
5. Descrivete cosa fa il programma se riceve da standard input 5.
6. Scrivete il valore di `os.Args[0]` se si eseguono da terminale i comandi:

```
go build tre_casi.go
./tre_casi 7 12 6
```

7. Qual è l'output del programma quando si eseguono da terminale i comandi precedenti e si inserisce 2 da standard input?
8. Riassumete il comportamento del programma con una frase del tipo: "Il programma legge ... da riga di comando e ... da standard input, quindi ... infine stampa su standard output ...". La frase deve tenere conto dei vari casi che si possono verificare.

3 Classificazione di triangoli

Considerate il seguente programma `triangoli.go`. Il programma riceve da standard input le lunghezze dei lati di un triangolo e classifica il triangolo come equilatero, isoscele o scaleno.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c int
7
8     fmt.Println("Inserisci tre numeri separati da spazi:")
9     fmt.Scan(&a, &b, &c)
10    if a == b && a == c {
11        fmt.Println("Il triangolo e' _____")
12    } else if a == b || a == c || b == c {
13        fmt.Println("Il triangolo e' _____")
14    } else {
15        fmt.Println("Il triangolo e' _____")
16    }
17 }
```

1. Completate le righe 11, 13 e 15 in modo che il programma stampi l'output corretto.
2. Modificate il programma affinché prima di procedere con la classificazione verifichi questa proprietà fondamentale: in un triangolo, ogni lato è più corto della somma degli altri due. Salvate il programma con nome `triangoli_bis.go` e caricatelo sul sito upload.

4 Numeri

Considerate il seguente programma `numeri.go`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     x := 0
8     for {
9         fmt.Scan(&n)
10        if n == 0 {
11            break
12        }
13        x += n
14    }
15    fmt.Println(x)
16 }
```

Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande

1. Cosa fa l'istruzione alla riga 11?
2. Cosa deve avvenire affinché il ciclo for termini?
3. Se viene eseguita l'istruzione alla riga 11, quale è la successiva riga di codice che viene eseguita?
4. Cosa si aspetta in input il programma?
5. Date un nome più significativo alla variabile `x`.
6. Il programma cosa fa di ciascun numero che riceve in input? Spiegate in dettaglio.
7. Riassumete il comportamento del programma con una frase del tipo: "Il programma legge da standard input ..., calcola ... e stampa ...".
8. Modificate il programma in modo che legga esattamente 10 numeri diversi da 0, stampi il risultato e termini. Salvate il nuovo programma con nome `numeri_bis.go` e caricatelo sul sito di upload.
9. Modificate il programma `numeri.go` in modo che legga da standard input una sequenza di interi terminata da 0 e calcoli la media dei numeri della sequenza. Salvate il nuovo programma con nome `numeri_ter.go` e caricatelo sul sito di upload. *Suggerimento: cosa deve fare il programma nel caso in cui l'input sia formato solo dal numero 0?*

5 Array

Considerate il seguente programma `array.go`. Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var s [10]int
7     var n int
8
9     fmt.Println("Inserisci 10 numeri:")
10    for i := 0; i < 10; i++ {
11        fmt.Scan(&s[i])
12    }
13    fmt.Println("Inserisci un numero:")
14    fmt.Scan(&n)
15
16    p := -1
17    for i, z := range s {
18        if z == n {
19            p = i
20        }
21    }
22    fmt.Println(p)
23 }
```

1. Cosa rappresentano le variabili `i` e `p` usate nel ciclo `for` nelle righe 17-21?
2. Riassumete con una frase cosa fa il programma.
3. Perché la variabile `p` è inizializzata a -1?
4. Scrivete un esempio di input su cui il programma stampa -1.
5. Scegliendo di scorrere l'array partendo dall'ultimo elemento invece che dal primo, come va modificato il programma? La variabile `p` e l'assegnamento in riga 19 sono ancora necessari? Salvate il programma con nome `array_bis.go` e caricatelo sul sito di upload.
6. La variabile `i` che visibilità (*scope*) ha nel programma `array.go`? E in `array_bis.go`?
7. Il ciclo `for` quante volte viene eseguito in `array.go`? E in `array_bis.go`? Spiegate.
8. Riprogettate il programma in modo che legga da riga di comando il numero da memorizzare nella variabile `n`. Salvate il programma con nome `array_ter.go` e caricatelo sul sito di upload. Suggestimenti: 1) potrebbe esservi utile la funzione `Atoi` del package `strconv`; 2) è necessario utilizzare un array e memorizzare tutti i numeri inseriti da standard input?

6 Matrice

Considerate il seguente programma `matrice.go`. Analizzate il codice e rispondete per iscritto alle seguenti domande

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     const R, C = 3, 4
7     var a [R][C]int
8
9     for i := 0; i < R; i++ {
10         for j := 0; j < C; j++ {
11             fmt.Scan(&a[i][j])
12         }
13     }
14
15     for i := 0; i < R; i++ {
16         for j := 0; j < C; j++ {
17             a[i][j] *= 2
18         }
19     }
20
21     fmt.Println(a)
22 }
```

1. Cosa indica la variabile `i`?
2. A cosa serve il ciclo alle righe 15-17?
3. Il ciclo `for` alle righe 15-17 può essere sostituito con un ciclo `for range`?
4. Riassumete in una frase cosa fa il programma.

7 Maiuscole

Il seguente programma `maiuscole.go` legge da standard input un testo ed emette in output il testo stesso, ma tutto in maiuscole. Analizzate il codice sorgente e rispondete per iscritto alle domande che seguono.

```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "os"
7 )
8
9 func main() {
10     var s string
11     const BASE_MINU = 'a'
12     const BASE_MAIU = 'A'
13
14     scanner := bufio.NewScanner(os.Stdin)
15     fmt.Println("Inserisci la stringa da mettere in maiuscolo:")
16     if scanner.Scan() {
17         s = scanner.Text()
18     }
19
20     fmt.Println("La stringa e' lunga", len(s))
21
22     for _, char := range s {
23         if char >= 'a' && char <= 'z' {
24             n := char - BASE_MINU
25             char = BASE_MAIU + n
26         }
27         fmt.Print(string(char))
28     }
29     fmt.Println()
30 }
```

1. Senza eseguire il programma, stabilite qual è l'output quando il programma riceve da standard input allo Zoo
2. A cosa serve l'underscore alla riga 22?
3. Alla riga 22 che valore assumerebbe `char` nel caso in cui l'underscore non fosse presente?
4. Cosa verifica la condizione dell'`if` alla riga 23?
5. Riscrivete la condizione dell'`if` di riga 23 usando un metodo del package `unicode`.
6. A cosa serve l'assegnamento nella riga 24?
7. Spiegate quale è il valore assunto da `char` in seguito all'assegnamento di riga 25.

8 Funzione

Considerate il seguente programma `funzione.go`. Analizzate il codice e rispondete per iscritto alle che seguono.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c float64
7     fmt.Print("Scrivi tre numeri: ")
8     fmt.Scan(&a, &b, &c)
9
10    media12 := media(a, b)
11    fmt.Println("media 1-2:", media12)
12
13    fmt.Println("media 2-3:", media(b, c))
14
15    somma := media(a, b) + media(b, c) + media(a, c)
16
17    fmt.Println("somma medie:", somma)
18
19 }
20
21 func media(n1, n2 float64) float64 {
22     return (n1 + n2) / 2
23 }
```

1. Quanti argomenti riceve la funzione `media`?
2. Di che tipo sono gli argomenti della funzione `media`?
3. Qual è il tipo del valore restituito dalla funzione `media`?
4. Indica le righe in cui la funzione `media` è invocata e, per ciascuna, come viene usato il valore restituito.

9 Cifre

Considerate il seguente programma `cifre.go`. Analizzate il codice e rispondete per iscritto alle che seguono.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, i int
7     var stop bool
8     var a [10]int
9
10    fmt.Print("Scrivi un numero intero >=0: ")
11    fmt.Scan(&n)
12
13    for n > 0 && !stop {
14        c := n % 10
15        n /= 10
16        fmt.Println(c)
17        a[i] = c
18        i++
19        if i >= 10 {
20            stop = true
21        }
22    }
23
24    if stop {
25        fmt.Println("numero troppo grande")
26    } else {
27        fmt.Println(a)
28    }
29
30 }
```

1. Che differenza c'è tra il ciclo `for` usato qui e quello utilizzato nell'esercizio 4?
2. A cosa serve la variabile `stop`?
3. Senza eseguire il programma stabilite qual è l'output quando il programma riceve da standard input 12345. E quando riceve 112233445566?
4. Riassumete il comportamento del programma con una frase del tipo: "Il programma legge . . . da standard input, quindi . . . , infine stampa su standard output . . . ". La frase deve tenere conto dei vari casi che si possono verificare.