



Didattica dell'Informatica  
a.a. 2019-20

Esercitazione sulla comprensione di programmi  
(nome: esComprensioneProgrammi)

**Consegna**

La tabella in Figura 1 illustra i diversi aspetti del processo di comprensione di un programma secondo un modello proposto da C. Schulte, detto *Block Model*.

A coppie svolgete l'esercizio 2 di comprensione di codice; dopodiché indicate, per ciascuna domanda, sia il suo scopo didattico, sia, facendo riferimento alla tabella, a quali aspetti del *Block Model* fa riferimento.

Per casa, per il 28/04/20, sempre lavorando nelle stesse coppie, scegliete uno tra i due esercizi 3 e 4 e svolgete lo stesso lavoro che avete fatto per l'esercizio 2.

Pensate poi a un esercizio di scrittura di codice che potreste dare ai vostri (ipotetici) alunni e trasformatelo in un esercizio di comprensione, preparando una scheda di lavoro (nome: schedaComprensioneProgrammi).

In particolare:

- svolgete in modo esemplare l'esercizio,
- sostituite i nomi di variabili, funzioni, ecc. con nomi non significativi,
- scegliete quattro posizioni diverse nel *Block Model*,
- formulate quattro domande di analisi e comprensione del codice che facciano riferimento ai quattro aspetti del *Block Model* scelti,
- preparate una scheda di lavoro con il codice e le quattro domande,
- svolgete l'esercizio compilando la scheda.

<b>(M) Macro structure</b>	Understanding the overall structure of the program text	Understanding the <i>algorithm</i> underlying a program	Understanding the goal/purpose of the program (in the context at hand)
<b>(R) Relationships</b>	Relations & references between blocks (e.g. method calls, object creation, data access...)	Sequence of method calls, <i>object sequence diagrams</i> .	Understanding how subgoals are related to goals, how function is achieved by subfunctions
<b>(B) Blocks (Chunks)</b>	<i>Regions of Interest</i> (ROI) that syntactically or semantically build a unit	Operations of a block, a method, or a ROI (chunk from a set of statements)	Understanding the function of a block, seen as a subgoal
<b>(A) Atoms</b>	Language elements	Operation of a statement	Function of a statement: its purpose can only be understood in a context
	<b>(T) Text Surface</b>	<b>(P) Program Execution</b>	<b>(F) Function/Purpose</b>
<b>Duality</b>	Architecture/Structure Dimensions		Relevance/Intention Dimension

Figura 1: Il *Block Model*

## 1 Programma brutto

La funzione seguente realizza in modo assai discutibile un calcolo che può essere descritto brevemente.

```
1 int func f(int x) {
2   int a, b, c;
3
4   while ( x > 0 ) {
5     a = x % 10;
6     b = 1 - a%2;
7     c += b;
8     x /= 10;
9   }
10  return c
11 }
```

Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande.

1. Senza eseguire il programma al computer, tracciatene l'esecuzione quando  $x$  è uguale a 1344.

a	b	c	x

2. Che dato viene salvato nella variabile  $a$ ?
3. La variabile  $b$  che valori può assumere? Spiegate.
4. Noto il valore di  $x$ , sappiamo quante volte verrà eseguito il ciclo `while`?
5. Date un nome più significativo alle variabili  $a$ ,  $b$  e  $c$ .
6. Riassumete con una frase cosa restituisce la funzione  $f$ .

## 2 Cifre

Considerate la seguente porzione di codice.

```
1 int main() {
2     int i, len;
3     char s[80];
4     int f[10] = {0};
5
6     scanf("%s", &s);
7
8     len = strlen(s);
9
10    for (i=0; i<len; i++) {
11        c = s[i];
12        if ( '0' <= c && c <= '9' )
13            f[ c - '0' ]++;
14    }
15 }
```

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. Perché si usa un ciclo `for`?
2. A cosa serve l'`if` alla riga 12?
3. Se `c` è una cifra, che cosa indica `c - '0'`?
4. Come descrivereste il valore `f[i]`?
5. Spiegate perché `f` è dichiarata con una lunghezza pari a 10.
6. Date un nome più significativo all'array `f`.
7. Riassumete in una frase cosa fa questa porzione di codice.

### 3 Divisioni

Considerate la seguente porzione di codice.

```
1  var n, b int
2
3  var c []int //slice, cioe' array dinamico
4
5  fmt.Scan(&n, &b) //lettura di n e b
6
7  if n == 0 {
8      fmt.Print(n)
9  } else {
10
11     for n > 0 {
12         c = append(c, n%b)
13         n = n / b
14     }
15
16     for i := len(c) - 1; i >= 0; i-- {
17         fmt.Print(c[i])
18     }
19 }
```

Analizzate il codice sorgente e rispondete per iscritto alle seguenti domande.

1. Spiegate cosa fa il ciclo nelle righe 11-14.
2. Date dei nomi più significativi alle variabili  $n$  e  $b$ .
3. Spiegate cosa fa il ciclo nelle righe 16-18.
4. Date un nome più significativo al programma.
5. Riassumete in una frase cosa fa il programma
6. Se  $b$  vale 10, descrivete il valore di  $c[2]$  in relazione all'intero  $n$
7. Come è dichiarata  $c$ ? Si potrebbe dichiarare  $c$  come un array? Quale sarebbe lo svantaggio rispetto alla dichiarazione attuale?

## 4 Quadrato magico

Considerate la porzione di codice seguente. Il codice serve a costruire un quadrato magico memorizzandolo in un array bidimensionale. Un quadrato magico è una disposizione dei numeri  $1, 2, \dots, n^2$  –con  $n$  dispari– tale che in ogni riga, in ogni colonna e nelle due diagonali la somma dei numeri sia la stessa.

```
1 var quadrato = [n][n]int{}
2 var inew, jnew int
3
4 i := 0
5 j := n / 2
6 nn := n * n
7
8 for k := 1; k <= nn; k++ {
9   quadrato[i][j] = k
10
11  if i == 0 {
12    inew = n - 1
13  } else {
14    inew = i - 1
15  }
16
17  if j == n-1 {
18    jnew = 0
19  } else {
20    jnew = j + 1
21  }
22
23  if quadrato[inew][jnew] == 0 {
24    i = inew
25    j = jnew
26  } else {
27    i++
28  }
29
30 }
```

Rispondete alle domande seguenti:

1. Come deve essere dichiarato  $n$ ? Che assunzioni ha senso fare sul valore di  $n$ ?
2. Spiegate come è usata la variabile  $k$  e cosa rappresenta.
3. Spiegate a parole quale valore viene assegnato a  $inew$  nel blocco `if-else` nelle righe 11-15
4. Spiegate qual è la funzione del blocco `if-else` nelle righe 23-28
5. Senza eseguire il programma al computer, tracciatene l'esecuzione quando  $n$  è pari a 5.
6. Verificate la correttezza della vostra risposta al punto precedente, completando il programma con la stampa del quadrato magico.
7. Descrivete a parole la strategia per costruire il quadrato magico che è implementata dalla suddetta porzione di codice. Iniziate con "Si parte mettendo il numero 1 al centro della prima riga ..."