

Corso di Ingegneria del Software

Prova scritta del 10 dicembre 2004

- 1) Descrivere il modello di sviluppo *a cascata*, illustrandone la struttura e descrivendo sinteticamente le attività associate alle varie fasi di cui si compone. Indicare inoltre le caratteristiche di *visibilità* di questo modello di sviluppo. [punti 5]
- 2) Descrivere cosa sono le tecniche di *fault avoidance*, *fault tolerance* e *fault detection*, utilizzabili per produrre sistemi software affidabili. [punti 4]
In quali modi è possibile realizzare software affidabile, evitando sin dall'inizio l'introduzione di errori nel sistema? Dare una breve spiegazione. [punti 4]
- 3) Descrivere le strategie di testing *Top-Down* e *Bottom-Up*, indicandone i vantaggi e i limiti. [punti 4]
- 4) Commentate la seguente affermazione: “se un progetto software è in ritardo, possiamo sempre aggiungere personale per accorciare i tempi”. Secondo voi è vera? Perché? Motivare le risposte e fornire un esempio. [punti 4]
- 5) Che cosa si intende con i termini *verifica* e *validazione* di sistemi software? Che vuol dire effettuarle in modo “statico” o “dinamico”? Discutere. [punti 4]

- 6) Si consideri il seguente frammento di codice

```
/*
 * Parametri in input:
 *   vettore v[] ordinato,
 *   dimensione dim del vettore (1 ≤ dim ≤ 10),
 *   valore val da cercare
 * Risultati:
 *   i tale che v[i] == val; -1 se val non è presente nel vettore
 */
INT v[1..10];      /* parametro in input */
INT dim;          /* parametro in input */
INT val;          /* parametro in input */
INT i;            /* risultato */
INT a,b;
BOOL f;

a:=1; b:=dim; f:=false;
WHILE ( (a <= b) AND ( NOT f ) ) DO BEGIN
  i := (a+b)/2;
  IF ( v[i]==val )
    f := true;
  ELSE BEGIN
    IF ( v[i]<val )
      b := i-1;
    ELSE
      a := i+1;
  END;
END;
IF ( NOT f )
  i := -1;
```

- a) Definire cos'è e a cosa serve la complessità ciclotomica di un programma. [punti 3]
- b) Calcolare la complessità ciclotomica del programma, indicando anche quali sono i cammini indipendenti del grafo di flusso. [punti 3]

Corso di Ingegneria del Software

Prova scritta del 10 dicembre 2004

- 1) Describe the *waterfall* software development model, indicating its structure and describing the activities associated with its different phases. How good is the *visibility* of this software development model? Why? [5 points]
- 2) Describe how the techniques of *fault avoidance*, *fault tolerance* and *fault detection*, work in producing reliable systems. [4 points]
How is it possible to produce reliable software systems by avoiding, from the beginning, the introduction of errors in the system? Give a brief explanation. [4 points]
- 3) Describe the testing strategies *Top-Down* and *Bottom-Up*, indicating advantages and disadvantages of them. [4 points]
- 4) Discuss the following statement: “if a software project is late, it is always possible to add manpower in order to complete it faster”. Is it correct? Why? Motivate your answers and give an example. [4 points]
- 5) What is the meaning of the terms *verification* and *validation*? What does it mean to perform them *statically* or *dynamically*? Discuss. [4 points]

- 6) Consider the following code fragment:

```
/*
 * Input parameters:
 *   sorted vector v[],
 *   dimension dim of the vector (1 ≤ dim ≤ 10)),
 *   integer val to look for
 * Result:
 *   i such that v[i] == val; -1 if val is not in the vector
 */
INT v[1..10];      /* input parameter */
INT dim;           /* input parameter */
INT val;           /* input parameter */
INT i;             /* result */
INT a,b;
BOOL f;

a:=1; b:=dim; f:=false;
WHILE ( (a <= b) AND ( NOT f ) ) DO BEGIN
  i := (a+b)/2;
  IF ( v[i]==val )
    f := true;
  ELSE BEGIN
    IF ( v[i]<val )
      b := i-1;
    ELSE
      a := i+1;
  END;
END;
IF ( NOT f )
  i := -1;
```

- a) Define what the cyclomatic complexity of a software program is, and what it can be used for. [3 points]
- b) Compute the cyclomatic complexity of the program above, showing the independent paths of the flow graph. [3 points]