

**Fondamenti di Informatica – A.A. 2016-2017**

Ingegneria Informatica e Ingegneria delle Telecomunicazioni

Prof. ssa Maristella Matera

Appello del 13/09/2017



Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Voto: \_\_\_\_/30

Quesito	1.1	1.2	2	3.1	3.2	3.3	Tot
Punteggio Max	3	2	5	4	6	10	30
Valutazione							

**Istruzioni:**

- Il tempo massimo a disposizione per svolgere la prova è di 2h.
- È vietato consultare appunti e utilizzare calcolatrici, telefoni, PC o qualsiasi dispositivo elettronico.
- Il voto minimo per superare la prova è 18.

**Quesito 1 (5 punti).**

1. **(3 punti)** Con riferimento alla codifica in **complemento a 2 su 8 bit**, indicare (giustificando brevemente le risposte) quali delle seguenti operazioni possono essere effettuate correttamente:

1. +127 -108
2. -70 -113
3. +123 +105
4. -18 +56

Mostrare in dettaglio come avviene il calcolo delle operazioni 1. e 2., evidenziando bit di riporto e di overflow eventualmente ottenuti.

2. **(2 punti)** Dato il numero 11000000111100000000000000000000 rappresentato secondo lo standard in virgola mobile IEEE 754 a 32 bit, determinare il corrispondente valore in base 10.

**Quesito 2 (5 punti).** Scrivere una funzione ricorsiva, avente come prototipo *int prodotto(int a[], int n, int target)*, che restituisce 1 se il prodotto dei primi *n* elementi dell'array *a* è uguale al valore intero *target*, 0 altrimenti. Per esempio, dati i seguenti valori:

$$a[] = \{4, 2, 2, 1, 5, 6\}, n=3, target=10$$

la funzione restituisce 0, poiché il prodotto dei primi 3 elementi è diverso da 10. Dati gli stessi valori dell'array, la funzione restituirebbe 1 per *target=16*.

**Quesito 3 (20 punti).** Scrivere un programma C, completo di dichiarazioni e opportunamente strutturato in funzioni, che determini se in un file sono contenute stringhe che sono anagrammi di una stringa data in input dall'utente. Gli anagrammi eventualmente individuati nel file devono poi essere memorizzati in un secondo file. N.B.: Una stringa è l'anagramma di un'altra se e solo se entrambe le stringhe contengono gli stessi caratteri, anche eventuali loro duplicazioni; i caratteri possono poi apparire nelle due stringhe in ordine differente. Per esempio, data la parola <'c','a','p','r','a'>, <'p','a','r','c','a'> è un suo anagramma, mentre non lo è <'p','a','n','c','a'>.

Il programma deve essere realizzato in modo modulare secondo i seguenti requisiti:

1. La funzione **main** deve ricevere come **argomenti della linea di comando** il nome dei due file utilizzati dal programma, quello da cui leggere le stringhe da analizzare e quello di output. Deve poi leggere in input dall'utente la stringa per la quale trovare gli anagrammi nel file **(4 punti)**.
2. Una funzione dovrà essere dedicata a leggere le stringhe dal file; si supponga che nel file le parole siano memorizzate una per riga e che la lunghezza massima di ogni parola sia di 10 caratteri **(6 punti)**.
3. Una funzione dovrà essere dedicata ad analizzare ogni singola parola per verificarne se è anagramma della parola specificata dall'utente. Se la parola letta è un anagramma, la funzione dovrà scriverla nel file di output. **(10 punti)**.

N.B: Nell'analisi degli anagrammi è possibile far uso di un array aggiuntivo. Se necessario, è possibile far uso anche di altre funzioni oltre a quelle richieste dai punti precedenti.

**Quesito Bonus (3 punti).** Si scriva un programma in Python che calcoli la somma dei primi 100 numeri naturali (da 0 a 500 escluso).