

Istruzioni:

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

Esercizio 1 - Codifica delle Informazioni, Aritmetica Binaria (4 punti)

Dati i numeri $X = +AF_{\text{hex}}$ (*esadecimale*) e $Y = -77_{\text{dec}}$ si indichi il minimo numero di bit sufficiente a rappresentarli entrambi in complemento a due, li si converta in complemento a due, e se ne calcoli la somma ($X + Y$), indicando se si genera riporto dalla colonna dei bit più significativi (sì/no) e se si verifica overflow (sì/no).

Esercizio 2 – File, stringhe, interi e linea di comando (12 punti)

Un file testuale contiene informazioni sulla provenienza dei partecipanti a un congresso scientifico: ogni riga, nel formato *cognome nazione*, riguarda un diverso partecipante. Dal file, esemplificato a lato, si vuole estrarre una struttura dati che a ciascuna nazione associi il numero di partecipanti da essa provenienti (nell'es.: *Italia: 2, Chile: 1, USA: 3, UK: 1*).

```
Rossi Italia
Sanchez Chile
Smith USA
Smith UK
Bianchi Italia
Smith USA
Koothrappali USA
```

- (a) Si codifichi in C una funzione ... provenienze(FILE * fp, ...) che riceve un **file** (già aperto in modalità *lettura*) e rende disponibile all'ambiente chiamante una opportuna struttura dati (da definire) che contiene le coppie *(nazione, numero_partecipanti)*. Si consideri, per semplicità, che il file sia sempre ben formato (ogni riga, cioè, ha sempre esattamente due stringhe, ciascuna di massimo 25 caratteri). [8 p.]
- (b) Si codifichi in C un programma (main) che, ricevuto **da linea di comando** il nome di un file come quello d'esempio, lo apre, invoca la funzione sopra definita, e stampa opportunamente su stdout le statistiche calcolate dall'invocazione. Si gestiscano gli eventuali errori nei parametri del main e nell'apertura del file. [4 p.]

Esempio di possibile output del programma:

```
Italia : 2
Chile : 1
USA : 3
UK : 1
```

Esercizio 3 - Analisi di stringhe, Funzioni Ricorsive (6 punti)

Si codifichi ricorsivamente in C una funzione `int distingui(char s[], char up[], char low[])` che riceve una stringa *s* (da analizzare) e due stringhe inizialmente vuote, ma già allocate e di dimensione pari a quella di *s*. La funzione dovrà (i) copiare nelle stringhe *up* e *low* i caratteri alfabetici rispettivamente maiuscoli e minuscoli di *s*, nello stesso ordine in cui compaiono in *s*, e (ii) restituire al chiamante il numero di caratteri non alfabetici di *s* (non copiati). Esempio: con *s[]* = "2Ew?jm7y9TR£" avremmo *up[]* = "ETR", *low[]* = "wjmy", valore restituito: 5 (caratteri 2 ? 7 9 e £). Per il punteggio pieno è necessario che il risultato sia individuato ricorsivamente. Una soluzione sostanzialmente iterativa varrà al più 4 punti.

Esercizio 4 - Liste dinamiche (8 punti)

```
typedef struct nd { int val;
                  struct nd * next; } Nodo;
typedef Nodo * List;
```

Si consideri una lista dinamica di interi definita come sopra. Si codifichi in C una funzione

```
void somma_triplette( List lis )
```

che riceve come parametro una lista *e*, a partire dall'inizio, la modifica nel seguente modo: per ciascuna tripletta (tre nodi consecutivi) modifica il valore del primo nodo, sostituendolo con la somma dei tre valori della tripletta, e rimuove i due nodi successivi.

Esempi: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow$ diventa $6 \rightarrow 15 \rightarrow 7 \rightarrow$ \rightarrow rimane \rightarrow
 $1 \rightarrow 2 \rightarrow 1 \rightarrow 8 \rightarrow 3 \rightarrow$ diventa $4 \rightarrow 8 \rightarrow 3 \rightarrow$ $1 \rightarrow$ rimane $1 \rightarrow$
 $1 \rightarrow 2 \rightarrow 1 \rightarrow 8 \rightarrow 3 \rightarrow 1 \rightarrow$ diventa $4 \rightarrow 12 \rightarrow$ $1 \rightarrow 2 \rightarrow$ rimane $1 \rightarrow 2 \rightarrow$