



Informatica A – a.a. 2010/2011 – 23/02/2011

Cognome _____ Matricola _ _____

Nome _____ Firma dello studente _____

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** (o ripudiate) con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **È possibile scrivere a matita** (e non occorre ricalcare al momento della consegna!).
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti**.
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile **ritirarsi senza penalità**.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: **2h15m**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1 (4 punti)

Esercizio 2 (7 punti)

Esercizio 3 (5 punti)

Esercizio 4 (4 punti)

Esercizio 5 (8 punti)

Totale: (completo 28 punti)

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni

(a) Si costruisca la tabella di verità della seguente espressione booleana in tre variabili, badando alla precedenza tra gli operatori logici. Eventualmente si aggiungano le parentesi (1 punto).

$$\mathbf{A \text{ and } ((B \text{ or not } C) \text{ or } (not A \text{ or } B))}$$

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = 47_{dec}$ e $B = -38_{dec}$, li si converta, se ne calcolino la somma ($A+B$) e la differenza ($A-B$) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow (2 punti).

(c) Si converta il numero 123.625 in virgola fissa e in virgola mobile ($r = m \times 2^n$, con m e n codificati in binario, sapendo che $1/2 = 0.5$, $1/4 = 0.25$, $1/8 = 0.125$, $1/16 = 0.0625$, $1/32 = 0.03125$, $1/64 = 0.015625$, e $1/128 = 0.0078125$) (1 punto)

Esercizio 2 - Interpretazione del codice C

Si consideri il seguente codice:

```
#include <stdio.h>
#include <string.h>

int f (char *str1, char *str2);

int main () {
    char a[10] = "barbara";
    char b[10] = "anna";
    int i;

    i = f(a, b);
    printf("Risposta 1: %d\n", i);

    strcpy (a,"casa");
    strcpy(b,"casa");
    i = f(a, b);
    printf("Risposta 2: %d\n", i);

    strcpy (a,"casa");
    strcpy(b,"casale");
    i = f(a, b);
    printf("Risposta 3: %d\n", i);

    strcpy (a,"");
    strcpy(b,"casale");
    i = f(a, b);
    printf("Risposta 4: %d\n", i);

    strcpy (a,"");
    strcpy(b,"");
    i = f(a, b);
    printf("Risposta 5: %d\n", i);
}

int f (char *str1, char *str2) {
    if ( (*str1) == '\0' && (*str2) != '\0') {return -1;}
    if ((*str1) != '\0' && (*str2) == '\0') {return 1;}
    if ((*str1) == '\0' && (*str2) == '\0') {return 0;}
    if ((*str1) < (*str2)){return -1;}
    if ((*str1) > (*str2)){return 1;}
    return f(str1+1, str2+1);
}
```

1. Spiegare cosa fa la funzione "f".
2. Spiegare come funziona la funzione "f". Quali sono i casi base e i passi induttivi, e perchè?
3. Spiegare il significato delle condizioni poste dentro gli if della funzione "f".
4. Spiegare il passaggio dei parametri alla funzione "f", sia da parte della funzione main che da parte della funzione "f" stessa.
5. Indicare chiaramente cosa viene stampato a video dalla funzione "main".

Esercizio 3 - DB

Sia data la seguente basi di dati per delle pizzerie in franchising a Milano:

PIZZERIA(Id_pizzeria, Nome, Indirizzo, Id_proprietario, aDomicilio)

DIPENDENTE(Id_dipendente, Id_ristorante, Nome, Cognome)

CONSEGNA(Id_consegna, Data_consegna, Id_ragazzo, IndirizzoConsegna, NPizze, Totale)

1. Scrivere la query SQL che estrae il numero di pizzerie che effettuano la consegna a domicilio.
2. Scrivere la query SQL che estrae l'identificativo e il nome dei ristoranti che hanno effettuato più di 1000 consegne nel 2010. Per ciascuno di questi ristoranti si dia anche il numero totale di pizze consegnate nel 2010.
3. Scrivere la query SQL che estrae l'identificativo, il nome e il cognome dei dipendenti che hanno effettuato consegne nel 2010, e per ciascuno si dia anche il numero medio di pizze effettuate per ciascuna consegna effettuate in quel periodo. Si consideri solo ragazzi che hanno consegnato almeno 200 pizze.

Esercizio 4 - Domanda di Teoria

Descrivere le varie modalità di passaggio dei parametri nelle funzioni C. Per ognuna si riporti un esempio.

Esercizio 5 - Ricorsione in C

Si consideri il seguente codice:

```
typedef struct nodo {
    int valore;
    int occorrenze;
    struct nodo *next;
} node;
typedef node *ptrNode;

ptrNode addToList(int n, ptrNode t);
void printList(ptrNode t);

int main () {
    int num[10] = {0,1,2,2,2,5,6,6,7,9};
    int i;
    int n;
    ptrNode testa = NULL;

    for (i=0; i<10; i++) {
        n=num[i];
        testa = addToList(n, testa);
    }
    printList(testa);
}
```

Il codice prende un array di 10 interi e ne produce una visualizzazione compatta. Prima viene creata una lista di nodi usando la funzione "addToList", e poi la lista viene stampata usando la funzione "printList".

La lista è fatta nel seguente modo. Ogni nodo contiene un valore letto dall'array e il numero di occorrenze di quel valore nell'array. Se un valore è ripetuto più volte nell'array, esso viene rappresentato da un solo nodo nella lista. Ad esempio, preso l'array definito nel codice della funzione "main", viene prodotta la seguente lista:

(0,1) -> (1, 1) -> (2, 3) -> (5,1) -> (6, 2) -> (7,1) -> (9,1) -> NULL

Il primo numero tra parentesi graffe rappresenta il valore, il secondo numero le sue occorrenze nell'array.

1. Implementare la funzione "printList" in maniera **NON RICORSIVA**. La funzione stampi la lista usando la notazione usata poc'anzi nell'esempio.
2. Implementare la funzione "addToList" in maniera **RICORSIVA**. La funzione deve incrementare il numero di occorrenze del valore "n", se esiste già un nodo per quel valore nella lista. Se il nodo non esiste, la funzione deve creare un nuovo nodo in fondo alla lista per quel valore, e impostare il numero di occorrenze a 1.