



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - a.a 2019/2020 - 14 Gennaio 2020

Cognome: _____ Matricola: _____
Nome: _____ Firma: _____

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** con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- 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: **2h30m**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	3 punti	_____
Esercizio 2	3 punti	_____
Esercizio 3	4 punti	_____
Esercizio 4	6 punti	_____
Esercizio 5	12 punti	_____
Totale(28)		_____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (3 punti)

- (a) Si costruisca la tabella di verità della seguente espressione booleana in tre variabili, badando alla precedenza tra gli operatori logici. Eventualmente si aggiungano parentesi. Non si accetteranno soluzioni senza il procedimento. (1 punto)

(A or B and not C) and C OR not B and C

Risposta:

A	B	C	OUT
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

OUT = AC + !BC

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -128dec e B = 63, 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. Non si accetteranno soluzioni senza il procedimento. (1 punto)

Risposta:

128d = 1000 0000b - 8bit

-128 cp2 = 1000 0000b - 8bit.

63d = 011 1111b - 6 bit + 1 segno

-63 cp2 = 100 0001b - cp2 7bit

A + B = -128 + 63 = -65 = 1011 1111b - No riporto perduto né overflow

A - B = -128 - 63 = -192 = [1] 0100 0001 - Riporto perduto con overflow

NOME e COGNOME: _____

8 bit va da -128 a 127

- (c) Si converta il numero 43.625 in virgola fissa e in virgola mobile con codifica IEEE 754, 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 $1/128 = 0.0078125$. Non si accetteranno soluzioni senza il procedimento. (1 punto)

Risposta:

6bit + segno

43d = 010 1011b

$0.5 + 0.125 = 0.101b$

43.625 = 0101011.101 j- Virgola fissa

segno = 0

mantissa = 1.01011101 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 010 1110 1000 0000 0000 0000

Esercizio 2 - Domanda di teoria (3 punti)

- (a) Spiegare cosa si intende per allocazione dinamica della memoria. Quando è necessaria? Riportare un esempio di allocazione dinamica (ANSI C) per un array con dimensione N (N letto mediante una scanf).

- (b) Cosa sono i dangling reference (puntatori ciondolanti)? Riportarne un esempio.

Risposta:

Esercizio 3 - Comprensione del Codice (4 punti)

Scrivere il valore di n e cosa stampa a video il seguente codice:

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

int doSomething(char *a, char b);
int main(void) {
    char *a;
    char b;

    int n = 10;
    a = (char*)malloc(sizeof(char)*n);

    strcpy(a, " "); //INSERIRE QUI IL CODICE PERSONA o LA MATRICOLA
    b = a[strlen(a)];
    b = (int)doSomething(a,b);
    printf("%s %c\n",a,(b+'a'));
    return 0;
}

char doSomethingElse(char *c, char d){
    if (*c == d)
        return 0;
    else
        return 1+doSomethingElse(++c, d);
}

int doSomething(char *a, char b){
    int n;
    int val = 0;
    int i;
    n = (int) doSomethingElse(a,b);
    for (i=1;*a!= b; a++, n--, i++){
        val += (*(a+n-i)-'0');
        printf("%c", (val%26)+'a');
    }
    printf("\n");
    return val%26;
}
```

n: _____

printf:

NOME e COGNOME: _____

Risposta:

Esercizio 4 - Matlab (6 punti)

Scrivere il codice Matlab che restituisca i valori richiesti.
Attenersi al numero massimo di righe di codice indicato.

- (a) Creare un matrice A di dimensioni righe e colonne differenti e casuali da 5 a 10 e contenente solo numeri 4; (max 1 riga) (1 punto)

Risposta:

```
A = 4*ones(randi([5 10]), randi([5 10]))
```

- (b) Inserire dopo la prima colonna, una nuova colonna contenente numeri casuali da 0 a NxM dove N numero di righe e M numero di colonne. (max 1 riga) (2 punti)

Risposta:

```
A= [A(:,1) randi([0 size(A,1)*size(A,2)], [size(A,1) 1]) A(:,2:end)]
```

- (c) Eliminare le righe con media inferiore a 6 (max 1 riga) (1 punto).

Risposta:

```
A(mean(A,2)<6,:) = []
```

- (d) Calcolare la somma totale di tutte le colonne pari (max 1 riga) (1 punto).

Risposta:

```
sum(sum(A(:,2:2:end)))
```

- (e) Scrivere una funzione che presa in ingresso la matrice A, crea una nuova matrice "quadrata", ovvero elimina le righe o le colonne in eccesso in modo da ottenere una matrice con numero righe = numero di colonne (max 8 righe compreso end) (1 punto).

Risposta:

```
function quadrata = test(B)  
i = min(size(B));
```



```
    for j=1:1:i
        for k=1:1:i
            quadrata(k,j) = B(k,j);
        end
    end
end

oppure

function quadrata = test(B)
    i = min(size(B));
    quadrata = B(1:i,1:i);
end
```

Esercizio 5 - Programmazione C (12 punti)

Si vuole creare un programma per la gestione della tombola. La tombola è caratterizzata attraverso una struttura dati DINAMICA contenente i numeri estratti. Ogni numero estratto è caratterizzato dal numero stesso e da un vettore di 5 elementi contenete le vincite per "ambo", "terno", "quaterna", "cinquina", "tombola").

Es. vettore = 1 0 2 1 0 Significa che con quel numero sono stati vinti: 1 ambo, 0 terni, 2 quaterne, 1 cinquina e 0 tombole.

Le cartelle della tombola sono composte da una matrice 5x3 di numeri interi.

Tramite un'altra struttura dati, ad ogni numero della tombola (da 1 a 90) è associato una descrizione estratta dalla smorfia napoletana.

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere una funzione RICORSIVA che calcoli quante vincite sono state fatte fino ad ora (2 punti)
3. Scrivere una funzione per l'inserimento in coda di un nuovo numero estratto. Il numero estratto deve essere casuale da 1 a 90 ma non ancora estratto. La funzione restituisce 0 se è stato possibile estrarre il numero, 1 altrimenti (sono già stati estratti 90 numeri). La funzione deve inoltre restituire il significato della smorfia relativo all'ultimo numero estratto. (4 punti)
4. Scrivere una funzione che presa in ingresso la lista di numeri estratti e una cartella della tombola, aggiorni l'ultimo elemento della lista con le eventuali vincite sulla cartella (si consideri solamente la vittoria più importante). (5 punti)

Risposta:

```

typedef struct node{
    int numero;
    int vincite[5];
    struct nodo *next;
} node;

typedef node* ptrNode;

char smorfia[90][100];
int cartella[3][5];
ptrNode tombola;

int contaVittorie(ptrNode lista){
    if (lista == NULL)
        return 0;
    int i, val=0;
    for (i=0; i<5; i++)
        val += lista->vincite[i];
    return val + contaVittorie(lista->next);
}

ptrNode inserisciInCoda(ptrNode lista, char smorfia[][100], char stringa[100], int *ris)
{
    int numero;
    int trovato;
    ptrNode testa = lista;
  
```

```

ptrNode temp;
int i = 0;
do{
    i++;
    trovato = 0;
    lista = testa;
    numero = rand() % 90 + 1;
    while ((lista!=NULL) && (trovato==0)){
        if (lista->numero==numero)
            trovato = 1;
    }
}while((trovato==1) && (i<90));
if (i>=90)
{
    *ris =1;
    return testa;
}

lista = testa;
strcpy(stringa,smorfia[numero]);
temp = (ptrNode)malloc(sizeof(node));
temp->numero = numero;
temp->next = NULL;

*ris = 0;

if (lista == NULL)
    return temp;
else
{
    while(lista->next!=NULL)
        lista = lista->next;
    lista->next = temp;
    return testa;
}
}

void aggiornaTombola(ptrNode tombola, int cartella[3][5])
{
    int i,j;
    int num=0;
    int max = 0;
    int cont[3];
    for (j=0; j<3; j++)
        cont[j] = 0;

    if (tombola==NULL)
        return;

    while (tombola->next!=NULL)
    {
        for (i=0; i<3; i++)
        {
            for (j=0; j<5; j++){
                if (tombola->numero == cartella[i][j]) {
                    cont[i]++;
                    break;
                }
            }
        }
    }
}

```

```
tombola = tombola->next;
}
if (cont[0]+cont[1]+cont[2]==15)
    tombola->vincite[4]++;
else{
    if ((cont[0]>cont[1]) && (cont[0]>cont[2]))
        max = cont[0];
    else if ((cont[1]>cont[0]) && (cont[1]>cont[2]))
        max = cont[1];
    else
        max = cont[2];

    if (max>=2)
    {
        tombola->vincite[max-2]++;
    }
}
}
```

NOME e COGNOME: _____

NOME e COGNOME: _____