



Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - a.a 2018/2019 - 19 Febbraio 2019

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	4 punti	_____
Esercizio 3	3 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 or B) or A

Risposta:

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

OUT = A+B+B!C

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = -23_{\text{dec}}$ e $B = 41$, 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:

-23d = 101001 - 5bit + 1 di segno = 6bit
 41d = 0101001b - 6 bit + 1 segno = 7bit
 cp2 = 1010111b - cp2 7bit

$A + B = -23 + 41 = 18 = 0010010$ - riporto perduto no overflow
 $A - B = -23 - 41 = -64 = [1]1000000b$ - Riporto perduto no overflow

NOME e COGNOME: _____

7 bit va da -64 a 63

- (c) Si converta il numero 47.375 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

47d = 0101111

$0.25 + 0.125 = 0.011$

37.59375 = 0101111.011 j- Virgola fissa

segno = 0

mantissa = 01.01111.011 non normalizzata

exp = 5 + 127 = 132d = 10000100

IEEE754

segno (1bit) = 0

esponente (8bit) = 10000100

mantissa (23bit) = 101 1110 1100 0000 0000 0000

Esercizio 2 - Domanda di teoria (4 punti)

(a) Che cosa si intende per puntatore? A cosa serve?

(b) Fare un esempio per dichiarare:

- un puntatore ad una lista. Specificare anche la struttura base della lista in modo che contenga solo un campo di tipo intero.

- un puntatore ad una variabile char

(c) Dato c, puntatore di tipo char:

- allocare la memoria necessaria per la variabile puntata da c (malloc)

- riempire la memoria puntata da c con il carattere 'a'

Esercizio 3 - Comprensione del Codice (3 punti)

Scrivere cosa stampa a video il seguente codice:

```
#include <stdio.h>
#include <string.h>
#define N 6

char doFirstThing(char *a);
int thenTheSecond(char a);

int main(void) {
    char *a;
    char c;
    int d;

    a = (char*)malloc(sizeof(char)*N);
    strcpy(a,"abcde");
    c = doFirstThing(a);

    printf("%c\n",c);

    d = thenTheSecond(*a);
    printf("%c\n",*a);

    printf("%d\n",d);

    return 0;
}

char doFirstThing(char *a){
    int i,med = 0, n=0;

    for (i=0; *a!='\0'; i++)
    {
        med = med + *a;
        a = a+1;
        n = n+1;
    }
    a = a-1;
    med = med / n;
    return med;
}

int thenTheSecond(char a){
    a = a - 'a';
    return a;
}
```

Risposta:

c
a
0

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 la matrice M con dimensione di righe e colonne casuali (diverse) da 5 a 10 contenente solo valori pari a 1; (max 1 riga) (1 punto)

Risposta:

```
M = ones(randi([5 10],[1 2]));
```

- (b) Inserire nelle righe dispari valori casuali da 1 a 10 (max 1 riga) (1 punto).

Risposta:

```
M(1:2:end,:) = randi([1,10],size(M(1:2:end,:)))
```

- (c) Sommare 1 alle celle della matrice che contengono numeri dispari (max 1 riga) (1 punto).

Risposta:

```
M(mod(M,2)==1) = M(mod(M,2)==1)+1
```

- (d) Calcolare la media delle celle con valore maggiore di 3 (max 1 riga) (1 punto).

Risposta:

```
mean (M(M>3))
```

- (e) Scrivere una funzione che presa in ingresso la matrice M, crei una nuova matrice "specchio" N con la prima colonna in ultima posizione, la seconda in penultima... (max 8 righe) (2 punto).

Risposta:

```
function [N] = funzione(M)
    for i=1:size(M,2)
        N(:,size(M,2)-i+1) = M(:,i);
    end
end
```

Esercizio 5 - Programmazione C (12 punti)

Si vuole progettare un software che permetta di registrare trasazioni di criptovalute. Ogni record del database costituisce una coin della criptovaluta ed è costituito da un identificativo univoco di 10 cifre , il valore in euro al quale è stato acquistato, il valore in euro al quale è stato venduto, il nome del servizio utilizzato per lo scambio. Ogni coin ha un solo proprietario che è identificato da nome, cognome, identificativo del blocco posseduto e anno di acquisto. Sia i coin che i proprietari sono definiti per mezzo di una lista.

Svolgere l'esercizio attenendosi a quanto richiesto. NON È RICHiesto SCRIVERE IL MAIN.

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)
2. Scrivere la funzione per inserire un nuovo coin ed un nuovo proprietario. L'id è calcolato in automatico partendo da 0 e aggiungendo 1 ad ogni nuovo nodo. (2 punti)
3. Scrivere la funzione che preso in ingresso un identificativo, restituisca la differenza tra prezzo di vendita e prezzo d'acquisto. Restituisce 0 se l'identificativo non esiste o la lista è vuota. (3 punti)
4. Scrivere la funzione che, preso in ingresso nome e cognome del proprietario, permette di calcolare il guadagno/perdita facendo la differenza tra prezzo d'acquisto e prezzo di vendita di tutti i coin posseduti. (3 punti)
5. Scrivere la funzione che preso in ingresso un anno, salva e restituisce un array (lunghezza massima 100) con i valori di vendita di tutte i blocchi che sono stati scambiati per mezzo della piattaforma "coin-base". (3 punti)

Risposta:

```
/*
 Si vuole progettare un software che permette di registrare trasazioni di criptovalute. Ogni record del database
 costituisce una coin della criptovaluta ed è costituito da un identificativo univoco di 10 cifre , il valore in euro
 al quale è stato acquistato, il valore in euro al quale è stato venduto, il nome del servizio utilizzato per lo
 scambio. Ogni coin ha un solo proprietario che è identificato da nome, cognome, identificativo del blocco
 posseduto e anno di acquisto. Sia i coin che i proprietari sono definiti per mezzo di una lista.
 Svolgere l'esercizio attenendosi a quanto richiesto. NON RICHiesto SCRIVERE IL MAIN.
 */

1. Definire le strutture dati necessari per la realizzazione del programma. (1 punto)

2. Scrivere la funzione per inserire un nuovo coin. L'id è calcolato in automatico partendo da 0 e aggiungendo 1
 ad ogni nuovo nodo. (2 punti)

3. Scrivere la funzione che preso in ingresso un identificativo, restituisca la differenza tra prezzo di vendita e
 prezzo d'acquisto. Restituisce 0 se l'identificativo non esiste o la lista è vuota. (3 punti)

4. Scrivere la funzione che, preso in ingresso nome e cognome del proprietario, permette di calcolare il guadagno/
 perdita facendo la differenza tra prezzo d'acquisto e prezzo di vendita di tutti i coin posseduti. (3 punti)

5. Scrivere la funzione che preso in ingresso un anno, salva e restituisce un array (lunghezza massima 100) con i
 valori di vendita di tutte i blocchi che sono stati scambiati per mezzo della piattaforma "coin-base". (3 punti)
 */

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

typedef struct coin{
    int id;
    float acquisto;
    float venduto;
    char nomeServizio[20];
    struct coin *next;
} coin;
```



```

typedef coin *p_coin;

typedef struct proprietari{
    char nome[20];
    char cognome[20];
    int id;
    int anno;
    struct proprietari *next;
} proprietari;

typedef proprietari *p_proprietari;

p_coin inserisciNode(p_coin lista, p_proprietari *prop, float acq, float ven, char nomeServizio[], char nomeProprietario[], float guadagnoSuVendita(p_coin lista, int id);
float guadagnoTotale(p_coin listaCoin, p_proprietari listaProp, char nome[], char cognome[]);
float creaArray (p_coin listaCoin, p_proprietari listaProp, int anno, float array[], int *);
void stampaListe(p_coin coins, p_proprietari props);

int main(int argc, const char * argv[]) {

    int c;

    float acq, ven;
    char nome[20],cognome[20], nomeServizio[20];
    int id, anno;
    float gain, media;
    float arrayVen[100];
    int k;

    p_coin coins = NULL;
    p_proprietari proprietari = NULL;

    do{
        printf("\n\nMENU\n");
        printf("1) Inserisci nuova coin\n");
        printf("2) Calcola guadagno per ID coin\n");
        printf("3) Calcola guadagno per nome e cognome\n");
        printf("4) Crea array coinbase\n");
        printf("5) Stampa liste\n");
        printf("6) ESCI\n\n");
        printf(">> ");
        scanf("%d",&c);

        switch (c) {
            case 1:
                printf("Prezzo acquisto: ");
                scanf("%f",&acq);
                printf("Prezzo vendita: ");
                scanf("%f",&ven);
                printf("Nome servizio: ");
                scanf("%s",nomeServizio);

                printf("Nome proprietario: ");
                scanf("%s",nome);
                printf("Cognome proprietario: ");
                scanf("%s",cognome);
                printf("Anno: ");
                scanf("%d",&anno);

```

```

        coins = inserisciNodo(coins,&proprietari,acq,ven,nomeServizio,nome,cognome,anno);
        break;

    case 2:
        printf("ID: ");
        scanf("%d",&id);
        gain = guadagnoSuVendita(coins, id);
        printf("Il guadagno su %d %.2f\n",id,gain);
        break;
    case 3:
        printf("Nome: ");
        scanf("%s",nome);
        printf("Cognome: ");
        scanf("%s",cognome);
        printf("Il guadagno totale dell'utente %s %s : %.f",nome,cognome,guadagnoTotale(coins, propri
        break;
    case 4:
        printf("Anno: ");
        scanf("%d",&anno);
        media = creaArray (coins, proprietari, anno, arrayVen, &k);
        printf("Media %.2f\n\n",media);
        for(int i=0; i<k; i++)
            printf("%d: %.2f\n",i,arrayVen[i]);
        break;
    case 5:
        stampaListe(coins, proprietari);
        break;
    default:
        break;
}
}while(c != 6);
return 0;
}

p_coin inserisciNodo(p_coin lista, p_proprietari *prop, float acq, float ven, char nomeServizio[], char nom
{
    p_coin temp;
    p_proprietari tempProp;

    temp = (p_coin)malloc(sizeof(coin));
    temp->acquisto = acq;
    temp->venduto = ven;
    strcpy(temp->nomeServizio, nomeServizio);
    if (lista == NULL)
        temp->id = 0;
    else
        temp->id = lista->id+1;
    temp->next = lista;
    lista = temp;

    tempProp = (p_proprietari)malloc(sizeof(proprietari));
    strcpy(tempProp->nome,nome);
    strcpy(tempProp->cognome,cognome);
    tempProp->id = lista->id;
    tempProp->anno = anno;
    if (*prop == NULL)
        tempProp->next = NULL;
    else
        tempProp->next = *prop;
    *prop = tempProp;
}

```

```

        return lista;
    }

    float guadagnoSuVendita(p_coin lista, int id){
        if (lista == NULL)
            return 0;
        else
        {
            if (lista->id == id)
                return (lista->venduto-lista->acquisto);
            else
                return guadagnoSuVendita(lista->next, id);
        }
    }

    float guadagnoTotale(p_coin listaCoin, p_proprietari listaProp, char nome[], char cognome[]){
        if (listaProp==NULL)
            return 0;
        else
        {
            if ((strcmp(listaProp->nome,nome)==0) && (strcmp(listaProp->cognome, cognome)==0)) //NOME E COGNOME
                return guadagnoSuVendita(listaCoin,listaProp->id) + guadagnoTotale(listaCoin, listaProp->next, n
            else
                return guadagnoTotale(listaCoin, listaProp->next, nome, cognome);
        }
    }

    float creaArray (p_coin listaCoin, p_proprietari listaProp, int anno, float array[], int *k){
        p_coin testaCoin = listaCoin;
        float med = 0;
        int i=0;
        if (listaProp == NULL)
            return 0;
        else{
            while(listaProp!=NULL)
            {
                if (listaProp->anno == anno)
                {
                    listaCoin = testaCoin;
                    while(listaCoin!=NULL)
                    {
                        if ((listaCoin->id == listaProp->id) && (strcmp(listaCoin->nomeServizio,"coinbase")==0))
                        {
                            array[i] = listaCoin->venduto;
                            med = med + array[i];
                            i++;
                            break;
                        }
                        else
                            listaCoin = listaCoin->next;
                    }
                }
                listaProp = listaProp->next;
            }
        }
        *k = i;
        med = med/i;
        return med;
    }
}

```

```
void stampaListe(p_coin coins, p_proprietari props){
    if (coins!=NULL)
    {
        printf("ID: %d\tAcquisto: %.2f\tVendita: %.2f\tServizio: %s\n", coins->id, coins->acquisto, coins->vendita, coins->servizio);
        stampaListe(coins->next, props);
    }
    else{
        if (props!=NULL)
        {
            printf("ID: %d\tNome: %s\tCognome: %s\tAnno: %d\n", props->id, props->nome, props->cognome, props->anno);
            stampaListe(coins, props->next);
        }
    }
}
```

NOME e COGNOME: _____

NOME e COGNOME: _____