Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

Informatica A - Prof. A. Fuggetta - a.a 2022/2023 - 25 gennaio 2023

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.**
- Scrivere nome e cognome su tutti i fogli. Non verranno corretti compiti non firmati o con nome illeggibile
- 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: **2h**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	**5 punti**	_____
Esercizio 2	**6 punti**	_____
Esercizio 3	**7 punti**	_____
Esercizio 4	**14 punti**	_____
Totale(32)		_____

NOME e COGNOME: _____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (5 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.
Scrivere l'espressione semplificata. Non si accetteranno soluzioni senza il procedimento. (2 punti)

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

Risposta:
A+BC

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = -41dec e B = 24dec 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. (2 punti)

Risposta:
A = -41 = 1010111b
B = 24 = 011000b
-B = -24 = 101000b

A+B
1010111
0011000
———
1101111
= -17
NO Overflow e No Riporto perduto

A-B

1010111
1101000

0111111

Riporto perduto e overflow

(c) Si converta il numero 7,3 in virgola fissa e in virgola mobile con codifica IEEE 754 con precisione singola. Non si accetteranno soluzioni senza il procedimento. (1 punti)

Risposta: 7 = 111
0.3 = $0.\overline{01001}$

Virgolafissa : 0111.01001...
Vrigolamobile :
S = 0
M = 11010011001100110011001
E = 127 + 2 = 129 = 10000001

NOME e COGNOME: _____

Esercizio 2 - Teoria (6 punti)

Segnare con una crocetta le risposte che si ritengono corrette. Per ogni domanda, possono essere presenti da 1 a 4 soluzioni corrette.

(a) L'intervallo di valori rappresentabili con 5 bit, utilizzando la rappresentazione in complemento a DUE è:

- ☐ da -15 a 16
- √ **da -16 a 15**
- ☐ da 0 a 15
- ☐ da -8 a 7

(b) Dato il seguente programma C:

```
#include <stdio.h>

void main(){
   int i,c;
   c=0; i=0;
   while (i<10) {
      c++;
      if ((c % 2) != 0)
         i++;
      else
         i = i+c;
   }
   printf("c=%d\n",c);
}
```

- ☐ 7
- ☐ il ciclo while non termina mai, quindi non viene visualizzato nulla
- ☐ 8
- √ **6**

(c) Qual e' la codifica in virgola fissa del numero 29,2 (8 bit parte intera, 8 bit parte frazionaria)

- ☐ 00010101,10011001
- √ **00011101,00110011**
- ☐ 11100001,00110011
- ☐ 00011101,00110000

(d) Indicare quale/i delle seguenti affermazioni è/sono una definizione di lista dinamica.

- ☐ Una lista dinamica in C è una struttura dati che può contenere solo elementi di tipo numerico.
- √ **Una lista dinamica in C può contenere uno o più campi contenenti informazioni di diverso tipo, e un un puntatore con il quale è legato all'elemento successivo nell'heap.**
- ☐ Una lista dinamica in C è una struttura dati che memorizza le informazioni sempre in modo ordinato.
- ☐ Una lista dinamica in C può contenere uno o più campi contenenti informazioni di diverso tipo, e un un puntatore con il quale è legato all'elemento successivo nello stack.

NOME e COGNOME: _____

(e) Dato il prototipo di funzione

char f(int *uno, int due);

e le dichiarazioni di variabili

char *x;
int y;

indicare quale/i delle seguenti invocazioni di funzione rispetta/no il prototipo.

- ☐ x = f(y, x)
- ☐ x=f(y,*y)
- √ **x=f(y,y)**
- ☐ *x = f(y, y)
- √ **y=f(y,x)**

(f) Si consideri il seguente programma.

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

void first(int* n) {
  if (*n % 2 == 0) {
       *n += 1;
  }
}

int second(float x) {
    if ((int)x%3 == 0) {
        return (int)sqrt(x);
    }
    return 0;
}


int main () {
    int a[] = {3, 2, 1, 4, 0};
    int i;
    float val;
    for (i=5; i>0; i--){
        first(&a[i-1]);
        i = second(a[i-1]);
    }
    printf ("%d", a[2]);
    return 0;
}
```

Cosa stampa la funzione printf?

- ☐ 2
- ☐ 3
- √ **1**
- ☐ Non stampa nulla, da errore

NOME e COGNOME: _____

(g) **Matlab (7 punti)**

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

1. Creare una matrice quadrata di dimensione casuale tra 3 e 10, e contenente numeri casuali da 1 a 100. (1 riga - 1 punto)

Risposta:

```
A = randi([1 100],randi([3 10]))
```

2. Dividere per due tutte le celle contenenti un numero dispari. (1 riga - 1 punto)

Risposta:

```
A(mod(A,2)==1)=A(mod(A,2)==1)/2
```

3. Cancellare le righe la cui somma è inferiore a 300 (1 riga - 1 punto)

Risposta:

```
A(sum(A,2)<300,:)=[]
```

4. Aggiungere una colonna dopo l'ultima colonna, contenente le radici quadrate delle medie delle righe (1 riga - 1 punti)

Risposta:

```
A = [A sqrt(mean(A,2))]
```

5. Creare un array contenente la diagonale della matrice (1 riga - 1 punti)

Risposta:

```
x = diag(A)
```

NOME e COGNOME: _____

6. Scrivere una funzione Matlab che presa in ingresso una matrice quadrata ed un numero n inferiore alla dimensione della matrice, costruisca una nuova matrice di dimensione nxn partendo dal centro della matrice. (2 punti)

Risposta:

```
function x = funzioncina(A,n)

   x = zeros(n);
        daY = ceil(size(A,1)/2)-floor(n/2);
        if (daY<0)
            daY=0;
        end
        aY = ceil(size(A,1)/2)+floor(n/2);
        if (aY>size(A,1))
            aY=size(A,1);
        end

        daX = ceil(size(A,2)/2)-floor(n/2);
        if (daX<0)
            daX=0;
        end
        aX = ceil(size(A,2)/2)+floor(n/2);
        if (aX>size(A,2))
            aX=size(A,2);
        end

        x = A(daY:aY,daX:aX)
end
```

NOME e COGNOME: _____

(h) **Programmazione C Liste (14 punti)**

Si ipotizzi di dovere realizzare un software per la gestione dei regali di natale indesiderati.
I regali indesiderati devono essere ri-venduti. Ogni regalo ha un nome, una descrizione, la persona da cui è stato regalato, un costo amazon, un prezzo di vendita, e una variabile che identifica se il regalo è stato venduto o meno.

Svolgere l'esercizio attenendosi a quanto richiesto. **NON E' RICHIESTO SCRIVERE IL MAIN**.

1. Si definiscano le strutture dati necessarie allo sviluppo di questo programma. (1 punto)

2. Si definiscano le dichiarazione nel main delle variabile di tipo lista e, per ognuno dei punti successivi, definire la chiamata alla funzione. (2 punti)

3. Si scriva la funzione ricorsiva per inserire un nuovo regalo in ordine di prezzo di acquisto.(4 punti)

4. Scrivere la funzione ricorsiva che permetta di calcolare quanti sono i regali venduti e l'ammontare totale guadagnato (somma totale degli importi a cui è stato venduto). (3 punti)

5. Scrivere una funzione che crei una nuova lista contenente solamente i regali venduti e li elimini dalla lista di partenza dei regali. (4 punti)

6. (SOLO PER CHI NON HA SUPERATO IL LABORATORIO)
Si definisca la funzione che presa in ingresso la lista ed il nome di un regalo, verifichi se nella lista sono presenti dei doppioni. La funzione restituisce 1 se il regalo è stato fatto più volte, zero se è presente una volta sola.

Risposta:

```
//
//  main.c
//  prova
//
//  Created by Paolo Perego on 22/06/23.
//

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

typedef struct regalo {
    char nome[100];
    char descrizione[200];
    char persona[20];
    float costo;
    float prezzo;
    int venduto;
} regalo;

typedef struct nodoRegalo{
    regalo gift;
    struct nodoRegalo *next;
} nodoRegalo;

typedef nodoRegalo* ptrRegalo;

ptrRegalo inserisciRegalo(ptrRegalo lista, regalo nuovoRegalo);
float calcolaTotale(ptrRegalo lista, int *numRegali);
ptrRegalo creaListaVenduti(ptrRegalo lista, ptrRegalo *venduti);
int trovaDoppioni(ptrRegalo lista, char nome[]);
void stampaLista(ptrRegalo lista);

int main(int argc, const char * argv[]) {
    ptrRegalo lista = NULL;
    ptrRegalo venduti = NULL;
    int ris;
    regalo gift;
    int num = 0;
    int doppi;
    float totale;

    do{
        printf("Cosa vuoi fare?\n");
        printf("1) Inserisci regalo\n");
        printf("2) Calcola regali venduti\n");
        printf("3) Crea lista\n");
        printf("4) Cerca doppioni\n");
        printf("5) Stampa lista\n");
        printf("6) ESCI\n");
        printf(">> ");
        scanf("%d",&ris);

        switch(ris){
            case 1:
                printf("Inserisci dati regalo:\n");
                fpurge(stdin);
                scanf("%s",gift.nome);
```

```
                fpurge(stdin);
                gets(gift.descrizione);
                scanf("%f",&(gift.costo));
                scanf("%f",&(gift.prezzo));
                scanf("%d",&gift.venduto);
                lista = inserisciRegalo(lista, gift);
                break;
            case 2:
                num = 0;
                totale = calcolaTotale(lista, &num);
                printf("Totale %.2f\n",totale);
                printf("Regali venduti %d\n",num);
                break;
            case 3:
                lista = creaListaVenduti(lista, &venduti);
                stampaLista(venduti);
                break;
            case 4:
                doppi = trovaDoppioni(lista, "lego");
                printf("Doppione %d",doppi);
                break;
            case 5:
                stampaLista(lista);
                break;

        }
    }while(ris!=6);

    return 0;
}

ptrRegalo inserisciRegalo(ptrRegalo lista, regalo nuovoRegalo){
    ptrRegalo nodo;
    if ((lista==NULL) || (lista->gift.costo>=nuovoRegalo.costo)){
        nodo = (ptrRegalo)malloc(sizeof(nodoRegalo));
        nodo->next = lista;
        nodo->gift = nuovoRegalo;
        return nodo;
    }

    lista->next = inserisciRegalo(lista->next, nuovoRegalo);
    return lista;
}

float calcolaTotale(ptrRegalo lista, int *numRegali){
    if (lista == NULL){
        return 0;
    }

    if (lista->gift.venduto == 1)
    {
        (*numRegali)++;
        return lista->gift.prezzo-lista->gift.costo + calcolaTotale(lista->next, numRegali);
    }
    else
        return calcolaTotale(lista->next, numRegali);
}

ptrRegalo creaListaVenduti(ptrRegalo lista, ptrRegalo *venduti){
    ptrRegalo temp;
```

```
    if (lista==NULL)
        return lista;

    if (lista->gift.venduto == 1)
    {
        *venduti = inserisciRegalo(*venduti, lista->gift);
        temp = lista;
        lista = lista->next;
        free(temp);
        lista = creaListaVenduti(lista, venduti);
    }
    else
        lista->next = creaListaVenduti(lista->next, venduti);
    return lista;
}

int trovaDoppioni(ptrRegalo lista, char nome[]){
    int trovato = 0;
    if (lista==0)
        return 0;

    for (; lista!=NULL; lista = lista->next){
        if (strcmp(lista->gift.nome,nome)==0)
            trovato++;
    }

    return (trovato>1);

}
void stampaLista(ptrRegalo lista){
    if (lista==NULL)
        return;
    printf("Nome: %s\n",lista->gift.nome);
    printf("Descrizione: %s\n",lista->gift.descrizione);
    printf("Costo: %.2f\n",lista->gift.costo);
    printf("Prezzo: %.2f\n",lista->gift.prezzo);
    printf("Venduto: %d\n",lista->gift.venduto);
    stampaLista(lista->next);
}
```