



Informatica – a.a. 2020/2021– 4° Appello – 11 Febbraio 2022

Cognome _____ **Matricola o Cod. Persona** _____

Nome _____ **Firma** _____

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine se necessario. **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** e qualsiasi **dispositivo elettronico**. Chi tenti di farlo vedrà **annullata** la sua prova.
- È **vietato** consultare **libri o appunti**. Chi tenti di farlo vedrà **annullata** la sua prova.
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- È possibile ritirarsi senza penalità.
- Tempo a disposizione: **2h:15**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1 (7 punti) _____

Esercizio 2 (10 punti) _____

Esercizio 3 (8 punti) _____

Esercizio 4 (8 punti) _____

Totale (33 punti) _____

Voto finale _____

Esercizio 1 Codifiche numeriche e algebra di Boole [7 punti]

(a) Si costruisca la tabella di verità della seguente espressione Booleana. [2 punti]

$$F(A,B,C) = \text{not (not C and not B) or A and C or not B and not A}$$

con altra notazione:

$$\overline{(\overline{C} \cdot \overline{B})} + A \cdot C + \overline{B} \cdot \overline{A}$$

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due entrambi i numeri $A = -44_{\text{dec}}$ e $B = 87_{\text{dec}}$, li si converta in complemento a due, se ne calcolino la somma $(A + B)$ e la differenza $(A - B)$ in complemento a due e si indichi se si genera riporto dalla colonna dei bit più significativi e se si verifica *overflow*. [4 punti]

(c) Si esprima il numero **duemilatrecentocinquantaquattro** (2354_{dec}) in base **undici** (cioè con cifre: 0,1,...,8,9,A) [1 punto]

Soluzione

A	B	C	$\overline{(\overline{C} \cdot \overline{B})} + A \cdot C + \overline{B} \cdot \overline{A}$	F(A,B,C)
0	0	0	$\text{not}(1 \cdot 1) + 0 \cdot 0 + 1 \cdot 1 = 1$	1
0	0	1	$\text{not}(0 \cdot 1) + 0 \cdot 0 + 1 \cdot 1 = 1$	1
0	1	0	$\text{not}(1 \cdot 0) + 0 \cdot 1 + 0 \cdot 1 = 1$	1
0	1	1	$\text{not}(0 \cdot 0) + 0 \cdot 1 + 0 \cdot 1 = 1$	1
1	0	0	$\text{not}(1 \cdot 1) + 1 \cdot 0 + 1 \cdot 0 = 0$	0
1	0	1	$\text{not}(0 \cdot 1) + 1 \cdot 0 + 1 \cdot 0 = 1$	1
1	1	0	$\text{not}(1 \cdot 0) + 1 \cdot 1 + 0 \cdot 0 = 1$	1
1	1	1	$\text{not}(0 \cdot 0) + 1 \cdot 1 + 0 \cdot 0 = 1$	1

Si osserva che nessuno dei due numeri A, B è una potenza di due positiva dunque è possibile applicare la formula:

$$n_{c2} = 1 + \max\{\text{ceil}(\log_2(44)), \text{ceil}(\log_2(87))\} = 1 + \max\{6, 7\} = 8 \text{ bit}$$

$$A = -44_{\text{dec}} = \text{not}(00101100_{c2}) + 1 = 11010011_{c2} + 1 = 11010100_{c2}$$

$$B = 87_{\text{dec}} = 01010111_{c2}$$

$$A + B = \dots = [1]00101011_{c2}$$

Si riporto oltre la cifra più significativa.

NO *overflow* (perché gli addendi hanno segno discorde)

$$A = -44_{\text{dec}} = 11010100_{c2}$$

$$-B = -87_{\text{dec}} = \text{not}(01010111_{c2}) + 1 = 10101000_{c2} + 1 = 10101001_{c2}$$

$$A + (-B) = \dots = [1]01111101_{c2}$$

Si riporto oltre la cifra più significativa.

SI *overflow* (perché gli addendi hanno

segno discorde da quello della somma)

$$2354_{\text{dec}} = 1850_{\text{undici}}$$

Se si continua sul retro di qualche foglio, indicare quale

Esercizio 2 (10 punti)

Definiamo la "somiglianza" S fra due stringhe come il rapporto tra il doppio del numero di "triplette di caratteri consecutivi" comuni ad entrambe le stringhe e la somma delle "triplette di caratteri consecutivi" in ciascuna stringa.

Nel caso in cui almeno una delle due stringhe sia composta da meno di 3 caratteri, la somiglianza è nulla ($S = 0.0$).

Scrivere un programma C, che acquisisca due stringhe da tastiera (*standard input*) e invocando un opportuno sottoprogramma che prende come parametri le due stringhe e restituisce la loro "somiglianza", riporti sullo schermo (*standard output*) quest'ultimo valore (calcolato conformemente alla definizione precedente).

Esempio:

string1: "ALEXANDRE"

string2: "ALEKSANDER"

triplette in comune: 2; "ALE", "AND"

Numero di triplette in string1: 7 (Oss.: num. caratteri di string1 meno 2)

Numero di triplette in string2: 8 (Oss.: num. caratteri di string2 meno 2)

somiglianza: $S = (2 \cdot 2) / (7 + 8) = 0.27$

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

#define MAXLUNG (50)

float somiglianza(char prima[], char seconda[]);

int main() {
    char p[MAXLUNG+1] = {'\0'}, s[MAXLUNG+1] = {'\0'};

    printf("\n Inserisci la 1ma stringa:");scanf("%f", p);
    printf("\n Inserisci la 2da stringa:");scanf("%f", s);
    printf("\n Somiglianza = %.2f", somiglianza(p,s));

    return 0;
} // end main

float somiglianza(char prima[], char seconda[]){
    float S = 0.0;
    int n1 = strlen(prima), n2 = strlen(seconda), cont = 0;
    if (n1 < 3 || n2 < 3) return S;
    for (int i = 2; i < n1; i++) {
        for (int j = 2; j < n2; j++) {
            if (prima[i] == seconda[j] && prima[i-1] == seconda[j-1]
                && prima[i-2] == seconda[j-2])
                cont += 1;
        } // end for j
    } // end for i
    S = (2*(float)cont) / (n1+n2-4);
    return S;
} // end somiglianza
```

Se si continua sul retro di qualche foglio, indicare quale

Esercizio 3 (7 punti)

Si consideri il seguente tipo di dato

```
typedef struct {
    char codice[5+1]; // codice univoco
    int  quantita;
    float prezzo;
    char descrizione[300];
} pezzoMeccanico_t;
```

Si scriva in C un sottoprogramma

```
...rimuoviDuplicati(...)
```

che riceva come parametri in ingresso un *array* con celle di tipo `pezzoMeccanico_t`, la lunghezza di tale *array* e **restituisca**:

- un *array* dinamico, il cui indirizzo di memoria iniziale è contenuto in una variabile di tipo `pezzoMeccanico_t*`, contenente una copia delle celle dell'*array* passato come parametro senza duplicati. *Quindi questo array dinamico avrà lunghezza minore o uguale a quella dell'array passato come parametro e quindi dovrà essere calcolata/stabilita.*
- la lunghezza del vettore dinamico (numero di elementi presenti nel vettore).

N.B.: è richiesta la **corretta** definizione dei parametri in ingresso e in uscita dal sottoprogramma.

```
pezzoMeccanico_t* rimuoviDuplicati(pezzoMeccanico_t A[], int lungIN,
                                   int*lungOUT) {
    pezzoMeccanico_t *temp, *res;
    temp = (pezzoMeccanico_t*) malloc(lung*sizeof(pezzoMeccanico_t));
    if (temp == NULL) { *lungOUT = 0; return NULL; }

    int k = 0, flag;
    for (int i = 0; i < lung; i++) {
        temp[k] = A[i];
        flag = 0;
        for (int j = i+1; j < lung && flag == 0; j++) {
            if (strcmp(A[i].codice, A[j].codice) == 0) flag = 1;
        } // end for j
        if (flag == 0) { k += 1; // se non ha duplicati che appaiono
                        // successivamente in elenco non
                        // sovrascriverò la cella k-esima
        }
    } // end for i

    res = (pezzoMeccanico_t*) malloc(k*sizeof(pezzoMeccanico_t));
    if (res == NULL) { *lungOUT = 0; return NULL; }

    for (int i = 0; i < k; i++) res[i] = temp[i];
    *lungOUT = k;

    free(temp);
    return res;
} // end rimuoviDuplicati
```

Se si continua sul retro di qualche foglio, indicare quale

Esercizio 4 (8 punti)

Si considerino **due** *file* di testo contenenti su ogni riga la descrizione di un pezzo meccanico con le stesse informazioni previste nella definizione del tipo di dato `pezzoMeccanico_t`, separate da un carattere di spazio (' '). Ogni riga è terminata dal carattere '\n'.

Esempio:

```
AMX72 128 20.75 mozzi
FBC56 79 15.7 flange
...
```

- (a) Scrivere un sotto-programma `...fondiFile (...)` che prenda come parametri i nomi di due *file* contenenti le informazioni di pezzi meccanici; le righe di ciascuno di essi sono ordinate in senso crescente secondo il codice di ogni pezzo. Il sotto-programma deve creare un terzo *file*, chiamato "PezziSenzaDuplicati.txt" con tutte le informazioni contenute nei due *file* i cui nomi sono passati come parametri.

Il terzo *file* (anch'esso con le righe ordinate in senso crescente secondo il codice di ogni pezzo) NON deve contenere righe con codici duplicati. Ogni sua riga deve contenere la descrizione di un pezzo meccanico che se presente anche in entrambi i *file* originali, esibisce come quantità la somma delle quantità presenti nei *file* originali e come prezzo il prezzo più alto tra i due riportati nei *file* originali.

[4 punti]

- (b) Scrivere un programma principale (`main`) che acquisito da tastiera il nome di un file contenente le informazioni su pezzi meccanici (organizzato come descritto in precedenza), riporti a video il prezzo medio dei pezzi meccanici disponibili nel maggior numero di pezzi (quantità). **[4 punti]**

```
#include <stdio.h>
#include <stdlib.h>
#define MAXLUNG (50)
typedef struct { char codice[5+1]; // codice univoco
                int quantita; float prezzo; char descrizione[300];
            } pezzoMeccanico_t;

int main() {
    char nome[MAXLUNG+1] = {'\0'};
    printf("\n Inserisci nome file:"); scanf("%s", nome);
    FILE *fp = fopen(nome, "r");

    if (fp == NULL) {
        fprintf(stderr, "\n Errore Apertura File! \n"); exit(-1);
    }
    pezzoMeccanico_t p;
    float accPrezzo = 0.0; int maxq = -1, u;
    while ( !feof(fp) ) {
        fscanf(fp, "%s %d %f %s\n", p.codice, p.quantita, p.prezzo,
            p.descrizione);

        if (p.quantita > maxq) {
            maxq = p.quantita; accPrezzo = p.prezzo; u = 1;
        }
        if (p.quantita == maxq) {
            accPrezzo += p.prezzo; u++;
        }
    } // end while
    fprintf(stdout, "\n il prezzo medio dei pezzi meccanici disponibili nel \
        maggior numero di pezzi e': %.2f", accPrezzo/(u*maxq));

    return 0;
} // end main
```

Se si continua sul retro di qualche foglio, indicare quale

```

void fondiFile(char nomePrimo[], char nomeSecondo[]) {
    // NB i file sono ordinati, per Hp, in ordine crescente di codice
    FILE *fpOut = fopen("PezziSenzaDuplicati.txt", "w"),
        *fp1 = fopen(nomePrimo, "r"),
        *fp2 = fopen(nomeSecondo, "r");

    if (fp == NULL || fp1 == NULL || fp2 == NULL) {
        fprintf(stderr, "\n Errore Apertura File! \n");
        exit(-1);
    }
    pezzoMeccanico_t pPrimo, pSec;
    fscanf(fp1, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
        pPrimo.prezzo, pPrimo.descrizione);
    fscanf(fp2, "%s %d %f %s\n", pSec.codice, pSec.quantita,
        pSec.prezzo, pSec.descrizione);
    while ( (!feof(fp1)) && (!feof(fp2)) ) {

        if (strcmp(pPrimo.codice, pSec.codice) < 0) {
            fprintf(fpOUT, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
                pPrimo.prezzo, pPrimo.descrizione);
            fscanf(fp1, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
                pPrimo.prezzo, pPrimo.descrizione);
        }
        else
        if (strcmp(pPrimo.codice, pSec.codice) > 0) {
            fprintf(fpOUT, "%s %d %f %s\n", pSec.codice, pSec.quantita,
                pSec.prezzo, pSec.descrizione);
            fscanf(fp1, "%s %d %f %s\n", pSec.codice, pSec.quantita,
                pSec.prezzo, pSec.descrizione);
        }

    } // end while
    while ( !feof(fp1) ) {
        fprintf(fpOUT, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
            pPrimo.prezzo, pPrimo.descrizione);
        fscanf(fp1, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
            pPrimo.prezzo, pPrimo.descrizione);
    } // end while
    while ( !feof(fp2) ) {
        fprintf(fpOUT, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
            pPrimo.prezzo, pPrimo.descrizione);
        fscanf(fp2, "%s %d %f %s\n", pPrimo.codice, pPrimo.quantita,
            pPrimo.prezzo, pPrimo.descrizione);
    } // end while

    fclose(fpOUT); fclose(fp1); fclose(fp2);
} // end fondiFile

```