



Informatica – a.a. 2021/2022– 1° Appello – 24 Giugno 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.
- 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:30**

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]

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

con altra notazione:

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

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due entrambi i numeri $A = -128_{\text{dec}}$ e $B = +7E_{\text{hex}}$, 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 decimale **trecentottantotto virgola zero settantasette** ($388,077_{\text{dec}}$) in base **tedici** (cioè con cifre: 0,1,...,8,9,A,B,C) mantenendo la notazione in virgola fissa con tre cifre per la parte frazionaria [1 punto]

Soluzione

(a)

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

- (b) $-128_{\text{dec}} = 10000000_{\text{c2}}$ 8bit
 $+7E_{\text{hex}} = 01111110_{\text{c2}}$ 8bit

$(A+B) = 11111110_{\text{c2}}$ No *overflow*. No riporto oltre la cifra più significativa.

$(A-B) = 00000010_{\text{c2}}$ Si *overflow*. Si riporto oltre la cifra più significativa.

- (d) $388,077_{\text{dec}}$

$$388_{\text{dec}} = 23B_{\text{tedici}}$$

$$0,077_{\text{dec}} = 0,100_{\text{tedici}}$$

$$388,077_{\text{dec}} = 23B,100_{\text{tedici}}$$

Se si continua sul retro di qualche foglio, indicare quale

Esercizio 2 (10 punti)

Si consideri una matrice con $R > 0$ righe e $C > 0$ colonne, contenente numeri *floating point*.

Tale matrice viene impiegata per modellare i rilevamenti di l'altitudine in un appezzamento di terreno: il numero in ogni cella della matrice indica dunque l'altezza media, rispetto al livello del mare, di un metro quadro di terreno.

Scrivere un sottoprogramma C con il seguente prototipo:

```
... maximumPath(double M[][C], int rig, int col, ... )
```

che date le coordinate di una cella della matrice $M[][]$ (avente `rig` righe e `col` colonne), calcoli il percorso **in salita più lungo e più ripido ad ogni passo** che è possibile compiere muovendosi su una cella adiacente a partire dalla cella data (il percorso è da intendersi come numero di celle visitate).

Si noti che prima di ogni passo, le possibili direzioni di spostamento su un'altra cella possono essere ≤ 8 , a seconda della posizione della cella di partenza nella matrice.

N.B.: è ammesso definire altri sottoprogrammi ausiliari (se lo si ritiene utile), di cui occorrerà esibire il codice.

Soluzione

```
// Sottoprogramma che trova la cella di valore massimo tra le celle adiacenti
// alla cella corrente. Restituisce la differenza tra le coordinate della cella
// con il valore massimo e quelle della cella corrente.
// Il sottoprogramma restituisce i due valori anzidetti usando due parametri
// in uscita -- facenti uso del passaggio parametri per indirizzo.
void trovaMassimo(double M[][C], int rig, int col, int x, int y,
                  int* deltaX, int* deltaY) {

    int massimo = M[x][y];
    *deltaX = 0; *deltaY = 0;
    for (int i = -1; i <= +1; i++) {
        for (int j = -1; j <= +1; j++) {
            if (!(x+i < 0 || x+i >= rig || y+j < 0 || y+j >= col)) {
                if (massimo < M[x+i][y+j]) { // viene calcolato il percorso in
                    massimo = M[x+i][y+j]; // salita piu' lungo che ad ogni
                    *deltaX = i; *deltaY = j; // passo e' anche il piu' ripido
                } // Con due valori massimi viene scelto
            } // l'ultimo trovato
        } // end for i
    } // end for j
} // end trovaMassimo

int maximumPath(double M[][C], int rig, int col, int x, int y) {

    if ( rig <= 0 || rig > R || col <= 0 || col > C ) {
        return -1; // sottomatrice rig x col non valida!
    } // end if
    if ( x < 0 || x >= rig || y < 0 || y >= col ) {
        return -1; // coordinate cella iniziale non valide!
    } // end if

    int dX, dY, numCelle = 1; // il percorso ritornato e' inteso come num. di celle

    trovaMassimo(M, rig, col, x, y, &dX, &dY);
    while (!(dX == 0 && dY == 0)) {
        x = x+dX;
        y = y+dY;
        numCelle += 1;
        trovaMassimo(M, rig, col, x, y, &dX, &dY);
    } // end while

    return numCelle;
} // end maximumPath
```

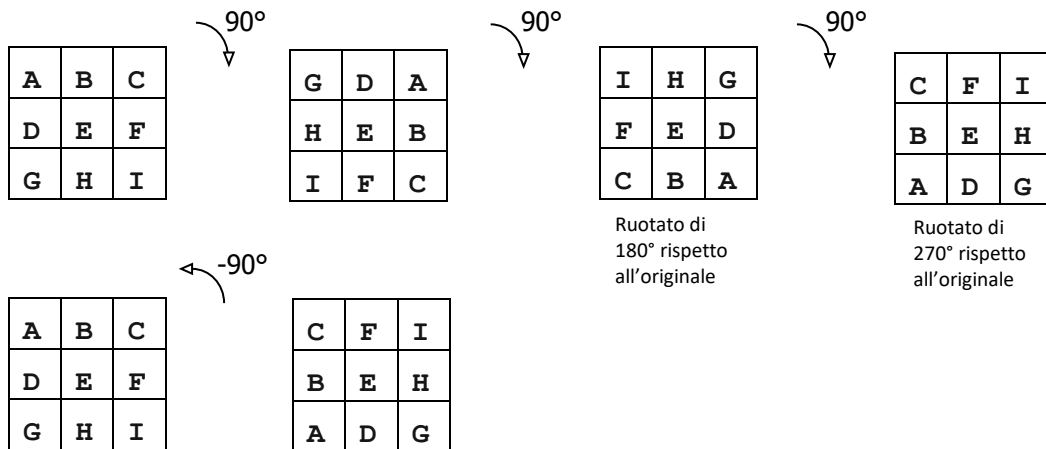
Se si continua sul retro di qualche foglio, indicare quale

Esercizio 3 (8 punti)

Si consideri una matrice quadrata di caratteri con numero di celle per lato $L > 0$.

Si scriva in C un sottoprogramma che abbia come parametro una matrice di caratteri, la lunghezza effettiva del suo lato e un **numero intero che indichi il numero di rotazioni di 90°** da poter far compiere alla matrice in senso orario, se il valore è positivo, oppure in senso antiorario, se il valore è negativo.

Lo scopo del sottoprogramma è quello di stampare a video il contenuto della matrice ruotata.



N.B.: potrebbe essere utile ricordare che l'operatore del linguaggio C, %, usato per il calcolo del resto della divisione intera, ha il seguente comportamento: $22 \% 12 == 10$; $-22 \% 12 == -10$.

Quindi nel caso in cui il numero di cui si vuole calcolare il resto fosse negativo e fosse utile avere il risultato dell'operazione espresso come un numero positivo il calcolo da fare sarebbe: $-22 \% 12 + 12 == 2$.

Se si continua sul retro di qualche foglio, indicare quale

Soluzione

```
#define L 50

void stampaRotazione(char mat[][L], int n, int numRotazioni) {

    if (n < 0 || n > L ) return;

    numRotazioni = numRotazioni % 4;
    if (numRotazioni < 0) numRotazioni += 4;
    // da questo punto in poi numRotazioni e' sicuramente in {0, +1, +2, +3}

    switch (numRotazioni) {
        case 0 : {
            for (int i = 0; i < n; i++) { // rig
                for (int j = 0; j < n; j++) { // col
                    printf("%c ", mat[i][j]);
                }
                printf("\n");
            }
            break;
        case 1 : {
            for (int j = 0; j < n; j++) { //rig
                for (int i = n-1; i >= 0; i--) { // col
                    printf("%c ", mat[i][j]);
                }
                printf("\n");
            }
            break;
        case 2 : {
            for (int i = n-1; i >= 0; i--) { // rig
                for (int j = n-1; j >= 0; j--) { // col
                    printf("%c ", mat[i][j]);
                }
                printf("\n");
            }
            break;
        default: { // case 3
            for (int j = n-1; j >= 0; j--) { // col
                for (int i = 0; i < n; i++) { // rig
                    printf("%c ", mat[i][j]);
                }
                printf("\n");
            }
            break;
        } // end switch
    } // end stampaRotazione
```

Se si continua sul retro di qualche foglio, indicare quale

Esercizio 4 (8 punti)

Si considerino le seguenti definizioni di tipo di dato

```
#define MAX_CONTATTI 100
#define MAX_NOME 30
#define MAX_TEL 11

typedef struct {
    char nome[MAX_NOME];
    char telefono[MAX_TEL];
} t_contatto;

typedef struct {
    t_contatto contatti[MAX_CONTATTI];
    int numero_contatti;
} t_rubrica;
```

- (a) Scrivere un sottoprogramma C che dati come parametri una variabile di tipo `t_rubrica` e il nome di un *file* di testo, memorizzi nella prima riga del *file* il numero totale di contatti e nelle righe successive tutti i contatti (uno per ciascuna riga del *file*).
- (b) Scrivere un sottoprogramma C che abbia come primo parametro il nome di un *file* di testo organizzato con le informazioni di una rubrica nello stesso modo usato dal sottoprogramma precedente, come secondo parametro un argomento di tipo `int` (che è assunto essere sempre un numero positivo con massimo tre cifre decimali) e che restituisca il numero di contatti il cui telefono inizia con tre caratteri numerici che corrispondono al valore intero passato come parametro.

Soluzione

```
// Restituisce 0 se tutto va a buon file, altrimenti restituisce -1.
int salvaSuFile(t_rubrica rub, char nomefile[]) {
    FILE *fp = fopen(nomefile, "w");
    if (fp == NULL) return -1;

    fprintf(fp, "%d\n", rub.numero_contatti);
    for (int i = 0; i < rub.numero_contatti; i++) {
        fprintf(fp, "%s %s\n", rub.contatti[i].nome, rub.contatti[i].telefono);
    } // end for

    fclose(fp);
    return 0;
} // end salvaSuFile

// Restituisce il numero di contatto con prefisso p, se tutto va a buon file;
// altrimenti restituisce -1.
int numPrefissi(char nomefile[], int p) {
    if ( !(p >= 100 && p <= 999) ) return -1;

    t_contatto contact;
    int totContatti = 0, num, count = 0;
    FILE *fp = fopen(nomefile, "r");

    if (fp == NULL) return -1;

    fscanf(fp, "%d\n", &totContatti);
    for (int i = 0; i < totContatti; i++) {
        fscanf(fp, "%s %s\n", contact.nome, contact.telefono);
        if (strlen(contact.telefono) >= 3) {
            num = (contact.telefono[2] - '0') + (contact.telefono[1] - '0') * 10 +
                (contact.telefono[0] - '0') * 100;
            if (num == p) count += 1;
        }
    } // end for
    fclose(fp);
    return count;
} // end numPrefissi
```

Se si continua sul retro di qualche foglio, indicare quale