



---

**Informatica – a.a. 2020/2021– 4° Appello – 24 Gennaio 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]

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

con altra notazione:

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

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due entrambi i numeri  $A = 96_{\text{dec}}$  e  $B = 57_{\text{dec}}$ , 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 **milleottocentoquarantacinque** ( $1845_{\text{dec}}$ ) in base  **tredici** (cioè con cifre: 0,1,...,8,9,A,B,C) [1 punto]

### Soluzione

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

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

$$n_{c2} = 1 + \max\{\text{ceil}(\log_2(96)), \text{ceil}(\log_2(57))\} = 1 + \max\{7, 6\} = 8 \text{ bit}$$

$$A = 96_{\text{dec}} = \dots = 01100000_{c2}$$

$$B = 57_{\text{dec}} = \dots = 00111001_{c2}$$

$$A + B = \dots = 10011001_{c2}$$

NO, riporto oltre la cifra più significativa.

SI, *overflow* (perché entrambi gli addendi hanno lo stesso segno e quest'ultimo è discorde con il segno del risultato)

¶

$$A = 96_{\text{dec}} = \dots = 01100000_{c2}$$

$$-B = -57_{\text{dec}} = \dots = 11000111_{c2}$$

$$A + (-B) = \dots = [1] 00100111_{c2}$$

SI, riporto oltre la cifra più significativa.

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

$$1845_{\text{dec}} = ABC_{\text{tredici}}$$

Se si continua sul retro di qualche foglio, indicare quale

## Esercizio 2 (10 punti)

Si consideri la seguente definizione del tipo di dato strutturato `ruota_t`

```
typedef struct {  
    char nomeRuota[10+1];  
    int numeri[5];  
} ruota_t;
```

per memorizzare nelle celle di una opportuna variabile di tipo *array* le estrazioni del gioco del lotto in una certa data, come mostrato nell'esempio seguente:

```
BARI 26 55 30 41 15  
CAGLIARI 83 44 43 29 37  
FIRENZE 35 65 61 2 47  
GENOVA 57 28 62 59 37  
MILANO 62 42 56 12 21  
NAPOLI 55 19 59 72 18  
PALERMO 48 34 62 41 55  
ROMA 81 53 50 27 51  
TORINO 40 84 38 64 35  
VENEZIA 5 28 52 3 53  
NAZIONALE 39 3 84 75 26
```

Si scriva un sottoprogramma C con il seguente prototipo:

```
int conteggioAssenti(ruota_t vett[11])
```

che restituisca il conteggio dei valori da 1 a 90 (quanti sono) che **NON** compaiono nell'insieme di estrazioni memorizzate nell'array passato come al argomento al sottoprogramma.

**Soluzione:**

```
int conteggioAssenti(ruota_t vett[11]) {  
    int cont = 0;  
    for (int num = 1; num <= 90; num++) {  
        flag = 1;  
        for (r = 0; (r < 11) && (flag == 1); r++) {  
            for (int j = 0; (j < 5) && (flag == 1); j++)  
                if (num == vett[r].numeri[j])  
                    flag = 0;  
        } // end for in r  
        if (flag == 1)  
            cont += 1;  
    }  
    return cont;  
} // end conteggioAssenti
```

*Se si continua sul retro di qualche foglio, indicare quale*

### Esercizio 3 (8 punti)

Si scriva in C un sottoprogramma

```
...ruotePesanti(...)
```

che riceva come parametri in ingresso un valore intero chiamato `soglia` e una variabile `vett` con 11 celle di tipo `ruota_t` (vedi def. nella traccia dell'esercizio 2) e **restituisca**:

- un vettore dinamico, il cui indirizzo di memoria iniziale è contenuto in una variabile di tipo `ruota_t*`, contenente una copia delle celle di `vett` in cui la somma dei numeri estratti supera il valore `soglia`. *Quindi questo vettore dinamico avrà lunghezza minore o uguale a 11 che 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 ed in uscita dal sottoprogramma.

#### Soluzione

```
ruota_t* ruotePesanti(int soglia, ruota_t vett[11], int* lung) {  
  
    int supera[11] = {0}, cont = 0;  
    /*  
    l'array supera[...] viene usato di seguito come array di valori 0 o 1:  
    se una sua cella conterra' 0 allora si sara' verificato che la cella  
    di vett[...] con lo stesso indice ha somma degli estratti che NON  
    supera la soglia.  
    Se una sua cella conterra' 1 allora si sara' verificato che la cella  
    di vett[...] con lo stesso indice ha somma degli estratti che supera  
    la soglia.  
    La variabile cont, conteggia il num. di celle non nulle in supera[...]  
    */  
    for (int i = 0; i < 11; i++) {  
        int sum = 0;  
        for (int j = 0; j < 5; j++) {  
            sum += vett[i].numeri[j];  
        }  
        if (sum > soglia) {  
            supera[i] = 1;  
            cont++;  
        }  
    } // end for  
  
    *lung = cont; // valore di ritorno passato per indirizzo  
    if (cont == 0) return NULL;  
  
    t_ruota* ret = (t_ruota*) malloc(cont * sizeof(t_ruota));  
    if (ret == NULL) return NULL;  
  
    int k = 0;  
    for (int i = 0; i < 11; i++) {  
        if (supera[i] == 1) {  
            ret[k] = vett[i];  
            k++;  
        }  
    } // end for  
  
    return ret;  
  
} // end ruotePesanti
```

Se si continua sul retro di qualche foglio, indicare quale

## Esercizio 4 (8 punti)

Nel gioco del Tris (*tic-tac-toe*) due giocatori scelgono a turno una casella vuota di una matrice 3×3 e vi inseriscono un loro segno identificativo: croce ('x') il primo giocatore e cerchio ('o') il secondo giocatore. Vince il gioco chi per primo riesce ad allineare tre dei suoi simboli in orizzontale, oppure in verticale oppure in diagonale. Se tutte le caselle della matrice sono state riempite senza realizzare un allineamento, il gioco finisce in parità.

- (a) Scrivere un sotto-programma `...calcola_vittoria(...)` che prenda come parametro una matrice di caratteri 3×3 contenente in ogni cella il carattere '-' per indicare che la stessa è vuota, oppure 'x', oppure 'o'. Il sotto-programma deve restituire il valore numerico 1 nel caso in cui la matrice indichi la vittoria del primo giocatore, 2 nel caso indichi la vittoria del secondo giocatore, 0 altrimenti. **[4 punti]**
- (b) Scrivere un programma principale (`main`) che acquisisca da un *file* di caratteri la matrice 3×3 corrispondente a una partita di Tris, invochi il sottoprogramma al punto (a), e stampi a video il valore da esso calcolato. Il *file* ha nome `partita.txt` e contiene tre righe di tre caratteri ciascuna per rappresentare la matrice di gioco; ogni riga è terminata dal carattere '\n'. **[4 punti]**

### Soluzione (a)

```
// per scelta implementativa non si considera il caso
// di partite impossibili con 2 vincitori
unsigned int calcola_vittoria(char mappa[3][3]) {
    int i, k;
    int num_x_in_diagonale1 = 0, num_x_in_diagonale2 = 0;
    int num_o_in_diagonale1 = 0, num_o_in_diagonale2 = 0;

    for (k = 0; k < 3; k++) {
        // controllo delle due diagonali
        if (mappa[k][k] == 'x') num_x_in_diagonale1++;
        else if (mappa[k][k] == 'o') num_o_in_diagonale1++;
        if (mappa[k][2-k] == 'x') num_x_in_diagonale2++;
        else if (mappa[k][2-k] == 'o') num_o_in_diagonale2++;

        // controllo della k-esima riga e della k-esima colonna
        int num_x_in_riga = 0, num_o_in_riga = 0;
        int num_x_in_colonna = 0, num_o_in_colonna = 0;

        for (i = 0; i < 3; i++) {
            if (mappa[k][i] == 'x') num_x_in_riga++;
            else if (mappa[k][i] == 'o') num_o_in_riga++;

            if (mappa[i][k] == 'x') num_x_in_colonna++;
            else if (mappa[i][k] == 'o') num_o_in_colonna++;
        } // end for i
        if (num_x_in_riga == 3 || num_x_in_colonna == 3) return 1;
        if (num_o_in_riga == 3 || num_o_in_colonna == 3) return 2;
    } // end for k

    if (num_x_in_diagonale1 == 3 || num_x_in_diagonale2 == 3) return 1;
    if (num_o_in_diagonale1 == 3 || num_o_in_diagonale2 == 3) return 2;
    return 0;
} // end calcola_vittoria
```

Se si continua sul retro di qualche foglio, indicare quale

## Soluzione (b)

```
#include <stdio.h> // per poter usare fopen(...), fscanf(...), printf(...)
#include <stdlib.h> // per poter usare exit(...)
```

```
unsigned int calcola_vittoria(char mappa[3][3]);
```

```
int main () {
```

```
    FILE *fp = fopen("partita.txt", "r");
    if (fp == NULL) {
        fprintf(stderr, "\nErrore nell'apertura del file ");
        fprintf(stderr, "- terminazione del programma! \n");
        exit(-1);
    }
```

```
    char mappa[3][3];
```

```
    int i;
    for (i = 0; i < 3; i++) {
        fscanf(fp, "%c", &mappa[i][0]);
        fscanf(fp, "%c", &mappa[i][1]);
        fscanf(fp, "%c", &mappa[i][2]);
        fscanf(fp, "\n");
    } // end for i
```

```
// codice alternativo
// per l'acquisizione dal file
int i, j;
char str[4+1]; // 3 char e un "a
               // capo",
               // + il '\0' aggiunto
               // dalla fgets...
for (i = 0; i < 3; i++) {
    fgets(str, 5, fp);
    for (j = 0; j < 3; j++) {
        mappa[i][j] = str[j];
    } // end for j
} // end for i
```

```
    printf("\nValutazione mappa del Tris:%u\n", calcola_vittoria(mappa));
```

```
    fclose(fp);
```

```
    return 0;
```

```
} // end main
```

```
// QUI andrebbe riportato il codice del sotto-programma calcola_vittoria
```

-----  
*Se si continua sul retro di qualche foglio, indicare quale*