

**PROVA SCRITTA DEL MODULO DI  
ELEMENTI DI INFORMATICA  
CORSO DI LAUREA IN INGEGNERIA BIOMEDICA  
24/2/2020**

**MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI**

**ESERCIZIO 1 (4 punti)**

Convertire in decimale il valore, espresso in binario, 10001.11.

**ESERCIZIO 2 (4 punti)**

Scrivere la tabella di verità dell'espressione booleana  $AB+B'CD+A'D'+C'$  (l'apice indica l'operatore NOT sulla variabile che lo precede).

**ESERCIZIO 3 (25 punti)**

(6 punti) Scrivere un programma in linguaggio C che, leggendo da un file "input.txt" una sequenza di valori reali, scriva in un file "output.txt" la media aritmetica dei valori progressivamente letti.

Esempio:

| Input.txt | Output.txt |
|-----------|------------|
| 12        | 12         |
| 6         | 9          |
| 0         | 6          |
| 2         | 5          |

Nello scrivere il programma si implementino ed utilizzino le seguenti strutture dati e funzioni:

(2 punti) Tipo di dato strutturato `tipo_dati` costituito da:

- un **float** `x` che conterrà l'ultimo valore letto dal file;
- un vettore statico di **float**, di cento posizioni, chiamato `media`, che contiene tutte le medie aritmetiche calcolate in corrispondenza dei valori letti. Il primo valore `media[0]` deve essere posto a 0;
- un intero `N` che contiene il numero di valori letti.

(3 punti) Funzione `inizializza_dati` con:

- parametro di ingresso: `void`;
- tipo in uscita: puntatore a `tipo_dati`.

La funzione inizializza lo slot `N` a 0 e il primo valore dello slot `media` a 0.0 di una variabile di tipo `tipo_dati` restituita in uscita.

(5 punti) Funzione `leggi_valore` con:

- parametri in ingresso:
  - puntatore a file chiamato `fp`
  - puntatore a `tipo_dati`
- parametro in uscita: intero

La funzione deve leggere un **float** da file `fp` e memorizzarlo nello slot `x` della variabile di tipo `tipo_dati`. Deve poi aggiornare lo slot `N` (numero di valori letti). Restituisce 0 se il file è finito.

(4 punti) Funzione `aggiorna_media` con:

- parametri: puntatore a variabile di tipo `tipo_dati`, chiamata `dati`
- tipo in uscita: **float**

La funzione deve calcolare la media aritmetica a partire dalla media presente nella posizione `N-1` dello slot `media` di `dati`, e dal valore letto nello slot `x` di `dati`, secondo la formula:

$$media(N) = \frac{(N-1) \cdot media(N-1) + x}{N}$$

(5 punti) Funzione `scrivi_media`, con

- parametri: puntatore a variabile di tipo `tipo_dati`, chiamata `dati`
- tipo in uscita: **void**

La funzione scrive su file "output.txt" la sequenza di **float** presente nello slot `media` di `dati`.

### Soluzione dell'esercizio 1

Per tradurre in decimale il valore espresso in virgola fissa 10001.11, dobbiamo applicare semplicemente l'algoritmo di conversione tenendo presente che le posizioni frazionarie sono numerate come -1 e -2. Si ottiene quindi:

$$(10001.11)_2 = 1 \cdot 2^4 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 16 + 1 + 0.5 + 0.25 = (17.75)_{10}$$

### Soluzione dell'esercizio 2

| A | B | C | D | AB | B'CD | A'D' | C' | Espr. |
|---|---|---|---|----|------|------|----|-------|
| 0 | 0 | 0 | 0 | 0  | 0    | 1    | 1  | 1     |
| 0 | 0 | 0 | 1 | 0  | 0    | 0    | 1  | 1     |
| 0 | 0 | 1 | 0 | 0  | 0    | 1    | 0  | 1     |
| 0 | 0 | 1 | 1 | 0  | 1    | 0    | 0  | 1     |
| 0 | 1 | 0 | 0 | 0  | 0    | 1    | 1  | 1     |
| 0 | 1 | 0 | 1 | 0  | 0    | 0    | 1  | 1     |
| 0 | 1 | 1 | 0 | 0  | 0    | 1    | 0  | 1     |
| 0 | 1 | 1 | 1 | 0  | 0    | 0    | 0  | 0     |
| 1 | 0 | 0 | 0 | 0  | 0    | 0    | 1  | 1     |
| 1 | 0 | 0 | 1 | 0  | 0    | 0    | 1  | 1     |
| 1 | 0 | 1 | 0 | 0  | 0    | 0    | 0  | 0     |
| 1 | 0 | 1 | 1 | 0  | 1    | 0    | 0  | 1     |
| 1 | 1 | 0 | 0 | 1  | 0    | 0    | 1  | 1     |
| 1 | 1 | 0 | 1 | 1  | 0    | 0    | 1  | 1     |
| 1 | 1 | 1 | 0 | 1  | 0    | 0    | 0  | 1     |
| 1 | 1 | 1 | 1 | 1  | 0    | 0    | 0  | 1     |

### Soluzione dell'esercizio 3.

Definizione del tipo di dato richiesto:

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

typedef struct
{
    float x;
    float media[100];
    int N;
} tipo_dati;

//Prototipi delle funzioni assegnate:

tipo_dati* inizializza_dati();
int leggi_valori(FILE *fp, tipo_dati* dati);
float aggiorna_media(tipo_dati* dati);
void scrivi_media(tipo_dati* dati);

int main()
{
    FILE *fp;
    tipo_dati* dati;
    float media;

    dati=inizializza_dati();

    fp=fopen("input.txt","r");

    while(leggi_valore(fp,dati))
    {
        media=aggiorna_media(dati);
        dati->media[dati->N]=media;
        printf("Media calcolata: %f\n",media); //bandierina
    }

    fclose(fp);

    scrivi_media(dati);

    return 0;
}

//Implementazione delle singole funzioni:

tipo_dati* inizializza_dati(void)
{
    tipo_dati *dati;
    dati=(tipo_dati*)malloc(sizeof(tipo_dati));
    dati->N=0;
    dati->media[0]=0.0;
    return dati;
}
```

```

int leggi_valore(FILE *fp, tipo_dati* dati)
{
    if (!feof(fp))
    {
        fscanf(fp,"%f",&(dati->x));
        (dati->N)++;
        //printf("Valore letto: %f %d\n",dati->x,dati->N); //bandierina
        return 1;
    }

    //printf("Uscita da leggi_valore: file finito\n");
    return 0;
}

float aggiorna_media(tipo_dati* dati)
{
    int N=dati->N;
    return ((N-1)*(dati->media[N-1])+dati->x)/N;
}

void scrivi_media(tipo_dati* dati)
{
    int i;
    FILE *fp;

    fp=fopen("output.txt","w");

    for(i=1; i<=dati->N; i++)
        fprintf(fp,"%f\n",dati->media[i]);

    fclose(fp);
}

```