



**Università degli Studi di Cagliari**

---

**FACOLTA' DI INGEGNERIA E ARCHITETTURA**  
**Tutorati Elementi di Informatica**

# **Esercitazione 2:**

## **Istruzioni base e Array**

### **Casi Pratici**

Elettrica, Elettronica ed Informatica

**Marco Melis**  
**marco.melis@unica.it**



### ■ Provateci voi:

- Scrivere un programma che identifichi le cifre ripetute all'interno di un intero di lunghezza qualsiasi e stampi una tabella che illustri i dati

■ Cifra:            0 1 2 3 4 5 6 7 8 9

3775576

■ Occorrenze: 0 0 0 1 0 2 1 3 0 0

### ■ Suggerimenti:

- Utilizzare **long int (%ld)** per interi molto grandi
- Utilizzare il carattere tabular '\t' per allineare
- $45 / 10 \rightarrow$  prima cifra     $45 \% 10 \rightarrow$  ultima cifra

```
1  #include <stdio.h>
2
3  int main(){
4
5      const int N_DIGITS = 10;
6      int cifra_vista[N_DIGITS];
7      long int numero;
8
9      printf("Inserire un intero: ");
10     scanf("%ld", &numero);
11
12     int i; // Inizializzo i contatori
13     for (i = 0; i < N_DIGITS; i++){
14         cifra_vista[i] = 0;
15     }
16
17     int cifra;
18     while(numero > 0){
19         cifra = numero % 10;
20         cifra_vista[cifra]++;
21         numero /= 10;
22     }
23
24     printf("Cifra:\t\t0 1 2 3 4 5 6 7 8 9\n");
25
26     printf("Occorrenze: ");
27     for (i = 0; i < N_DIGITS; i++){
28         printf("%d ", cifra_vista[i]);
29     }
30
31     return 0;
32 }
```

- Limitare l'uso della scanf a singole parole
- Nel caso di stringhe dal contenuto variabile ->
- Prende da *stdin* un carattere per volta

```
char nome[20];  
char ch;  
  
printf("Inserisci il nome ");  
int i = 0;  
while(i < 20){  
    ch = getchar();  
    if(ch == '\n'){  
        break;  
    }  
    nome[i++] = ch;  
}  
printf("%s", nome);
```

- Provateci voi:
  - Uno dei modi per aumentare l'efficacia di una password alfabetica è quello di sostituire alle vocali dei numeri, così da renderla alfanumerica
  - Scrivere un programma che prenda in ingresso una stringa e stampi la corrispondente password “complessa” sostituendo alle vocali dei numeri e aggiungendo un punto “.” ogni 5 caratteri.
    - A -> 4, E -> 3, I -> 1, O -> 0, U -> 6
- Suggerimenti:
  - Utilizzare **getchar** (non scanf) per ottenere i caratteri da tastiera
  - Gestire la sequenza in uscita come singoli char
  - Inserire una dimensione massima e minima della password originale

```
1  #include <stdio.h>
2
3  int main(){
4
5      const int dim = 20;
6      char seq[dim];
7
8      printf("Inserire la password (min 5 char, max %d char): ", dim);
9
10     char ch;
11     int i;
12     for(i = 0; i < dim - 1; i++){
13         ch = getchar();
14         if (ch == '\n'){
15             break;
16         }
17         seq[i] = ch;
18     }
19     seq[i] = '\0';
20
21     if(i < 5){
22         printf("La password deve essere lunga almeno 5 char\n");
23         return 0;
24     }
```

```
26   for(i = 0; i < dim; i++){
27       ch = seq[i];
28       if(ch == '\\0'){
29           break;
30       }
31       switch(ch) {
32           case 'a':
33               printf("4");
34               break;
35           case 'e':
36               printf("3");
37               break;
38           case 'i':
39               printf("1");
40               break;
```

```
41
42           case 'o':
43               printf("0");
44               break;
45           case 'u':
46               printf("6");
47               break;
48           default:
49               printf("%c", ch);
50       }
51       if ((i+1) % 5 == 0){
52           printf(".");
53       }
54   }
55   return 0;
56 }
57
```

### ■ Provateci voi:

- Scrivere un programma C che converta un numero binario di N bit (N inserito da tastiera) in un numero decimale
  - Il numero andrà inserito un bit alla volta, partendo dal bit meno significativo
  - Controllo sull'input: i dati inseriti devono essere interi e devono essere compresi nelle cifre della base
  - Stampare a schermo il risultato finale

### ■ Suggerimenti:

- Utilizzare **#define** per generalizzare il numero di cifre della base
- La *scanf* ritorna 1 se l'utente inserisce un numero



# Tipi/Espressioni – Controllo – Iterazione

## Conversione Binario -> Decimale

```
1  #include<stdio.h>
2  #define BASE 2
3
4
5  int main(){
6
7      int N; // numero di cifre che l'utente deve inserire
8      int peso = 0; // peso della cifra binaria
9      int bit;
10     int potenza = 1; // peso=0 ==> potenza 2^0=1
11     int numero_dec = 0;
12
13     printf("Questo programma converte un numero in base "
14           "%d ad un numero decimale.\n\n", BASE);
15
16     printf("Inserire il numero di cifre del numero da convertire: ");
17     while (scanf("%d", &N) != 1){
18         printf("Input non valido... inserire un intero: ");
19         /* rimuovo eventuali caratteri rimasti nel buffer fino al newline */
20         while (getchar() != '\n');
21     }
```



```
23 printf("\nInserire il numero partendo dalla cifra meno significativa...\n");
24
25 while(peso < N){
26     printf("Inserire la cifra %d^%d: ", BASE, peso);
27     scanf("%d",&bit);
28
29     if(bit >= 0 && bit < BASE){ // controllo se input e' corretto
30         // calcolo il valore decimale della cifra e lo sommo al valore precedente
31         numero_dec = numero_dec + bit * potenza;
32         peso = peso + 1;
33         potenza = potenza * BASE;
34     }else{
35         // controllo sulla cifra inserita, deve essere minore della base
36         printf("Cifra inserita non valida. Reinserire cifra.\n");
37     }
38 }
39
40 printf("\n\nIl valore inserito convertito in base 10 è: %d\n", numero_dec);
41
42 return 0;
43 }
```