

## [T01] Esercitazione del 25 febbraio 2022

---

### Istruzioni per l'esercitazione:

---

- Aprite il [form di consegna](#) in un browser e loggatevi con le vostre credenziali uniroma1.
- Scaricate e decomprimate sulla scrivania il [codice dell'esercitazione](#). Vi sarà una sotto-directory separata per ciascun esercizio di programmazione. Non modificate in alcun modo i programmi di test \*\_main.c.
- Rinominare la directory chiamandola cognome.nome. Sulle postazioni del laboratorio sarà /home/biar/Desktop/cognome.nome/.
- È possibile consultare appunti/libri e il materiale didattico online.
- Rispondete alle domande online sul modulo di consegna.
- **Finiti gli esercizi**, e non più tardi della fine della lezione:
  - **zippate la directory di lavoro** in cognome.nome.zip (`zip -r cognome.nome.zip cognome.nome/`).
- **Per consegnare:**
  - inserite nel form di consegna come autovalutazione il punteggio di ciascuno dei test forniti (inserite zero se l'esercizio non è stato svolto, non compila, o da errore di esecuzione).
  - fate **upload** del file cognome.nome.zip.
  - **importante:** fate logout dal vostro account Google!
- **Se avete domande** accedete a [Zoom](#) agli orari stabiliti per l'esercitazione, accedendo con la **mail istituzionale** uniroma1.it dello studente. Troverete online i docenti ed il tutor del corso. In alternativa, scrivete via mail ai docenti.

Per maggiori informazioni fate riferimento al [regolamento delle esercitazioni](#).

### Esercizio 1 (Palestra C)

---

Scrivere una funzione C uint2bin con il seguente prototipo

```
void uint2bin(unsigned x, char bin[32]);
```

che, dato un intero x senza segno a 32 bit e un buffer bin di 32 caratteri, ottiene la rappresentazione binaria del numero con il bit più significativo per primo. Gli 0 e 1 nel risultato devono essere rappresentati mediante i caratteri ASCII '0' e '1'.

Ad esempio, invocando la funzione con 0x0F0F0F0F, il buffer di output sarà "00001111000011110000111100001111".

Usare il main di prova nella directory di lavoro E1 compilando con `gcc e1_main.c e1.c -o e1`.

### Esercizio 2 (Palestra C)

---

Scrivere nel file E2/e2.c la vostra versione personale della funzione della libreria standard libc [strcpy](#) che copia la stringa str nel buffer dest e restituisce dest. Il prototipo della funzione da realizzare è il seguente:

```
char *my_strcpy(char *dest, const char *src);
```

Usare il main di prova nella directory di lavoro E4 compilando con `gcc e2_main.c e2.c -o e2`.

## Domande

---

Rispondi alle seguenti domande, tenendo conto che una risposta corretta vale 1 punto, mentre una risposta errata vale 0 punti.

**Domanda 1.** Un circuito combinatorio è:

- A. costruito mediante porte logiche e calcola funzioni booleane
- B. alla base della costruzione delle porte logiche
- C. un circuito che consente di mantenere uno stato ed è pertanto un possibile ingrediente per costruire memorie
- D. un circuito per la temporizzazione degli eventi in un sistema di calcolo

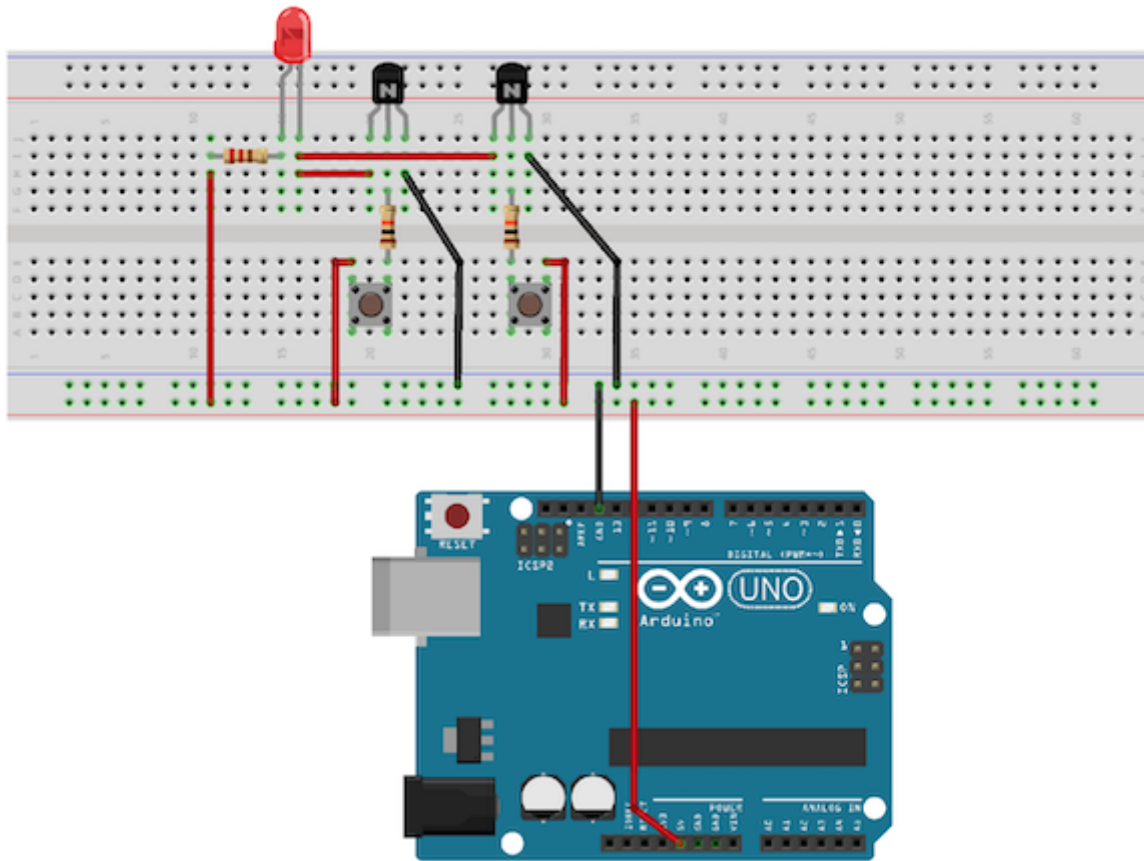
**Domanda 2.** Data una tensione da 12V e un LED che si illumina in presenza di una corrente nel range [50, 100] mA (oltre tale valore il LED si brucia) quale tra queste resistenze può essere inserita nel circuito per far illuminare il LED?

- A. 80
- B. 100
- C. 220
- D. 450
- E. 500

**Domanda 3.** Quale tra queste affermazioni su un flip-flop SR (latch) è falsa:

- A. può essere utilizzato per memorizzare un bit di informazione
- B. la configurazione  $S=0$  e  $R=1$  non garantisce  $Q = \text{NOT } Q'$
- C. la configurazione  $S=1$  e  $R=1$  potrebbe portare ad uno stato indefinito
- D. il flip flop non mantiene il suo stato quando non è alimentato (no corrente elettrica)

**Domanda 4.** Cosa calcola la seguente breadboard?



- A. NOT
- B. AND
- C. OR
- D. XOR
- E. Nessuna delle precedenti

## Soluzioni

### Esercizio 1 (Palestra C)

```
#include <string.h>
#include "e4.h"

void uint2bin(unsigned x, char bin[32]) {
    int idx;
    for (idx = 31; idx >= 0; idx--) {
        bin[idx] = '0' + (x & 1);
        x = x >> 1;
    }
}
```

### Esercizio 2 (Palestra C)

```
char *my_strcpy(char *dest, const char *src) {
    char* aux = dest;
    while(*src) *dest++ = *src++;
    *dest = '\0';
}
```

```
    return aux;  
}
```

## Domande

---

1. A - costruito mediante porte logiche e calcola funzioni booleane
2. C - 220
3. B - la configurazione  $S=0$  e  $R=1$  non garantisce  $Q = \text{NOT } Q'$
4. C - OR