

Matr.: \_\_\_\_\_ Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Architettura degli Elaboratori  
prova scritta di Assembly

ISTRUZIONI:

1. Scrivete subito il vostro nome, cognome e numero di matricola su questo foglio.
2. Aprite un editor a vostra scelta, scrivete una prima linea di commento con le stesse informazioni formattate nel seguente modo: *#matricola nome cognome*  
poi salvate il file chiamandolo: *cognomen.s* (con il vostro cognome e iniziale del nome!).
3. Tutti gli esercizi vanno scritti su questo file.
4. Per consegnare, bisogna risolvere i primi due esercizi e verificare con `xspim` che la soluzione sia corretta con i dati dell'esempio.
5. L'etichetta **main** va messa davanti alla funzione di test dell'esercizio 4 se ci arrivate e funziona altrimenti dovete metterla davanti al primo test (esercizio 2).
6. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
7. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

**Esercizio 1 (3 punti)**

Scrivere una funzione **subCopy** che prende come parametri un puntatore ad una stringa, due indici interi a 8bit senza segno e un puntatore ad un buffer. La funzione copia nel buffer la parte della stringa a partire dal primo indice fino al secondo escluso. Il buffer deve essere zero terminato. Si assuma che gli indici sono interni alla stringa. Ad esempio, data la stringa: "abcd\*str\*efghi", 5, 8; la funzione scrive nel buffer: "str".

**Esercizio 2 (1 punto)**

Scrivere una funzione **subCopyTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **subCopy** ai dati e stampa il buffer.

**Esercizio 3 (3 punti)**

Scrivere una funzione **copyDStr** con tre parametri: un puntatore ad una stringa, un delimitatore (carattere) e un buffer. La funzione trova gli indici delle prime due occorrenze del delimitatore nella stringa e chiama la funzione **subCopy** passandogli la stringa originale, i due indici trovati e il buffer. Si assuma che esistano i due delimitatori nella stringa. Ad esempio, data la stringa: "abcd\*star\*efghi" e il carattere '\*'; alla fine il buffer contiene: "star".

**Esercizio 4 (1 punto)**

Scrivere una funzione **copyDStrTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **copyDStr** ai dati e stampa il buffer.

NB. Si ricorda che per stampare un intero lo si deve caricare in `$a0`, caricare 1 in `$v0` e chiamare `syscall`; invece per stampare una stringa la si deve caricare in `$a0`, caricare 4 in `$v0` e chiamare `syscall`.

Matr.: \_\_\_\_\_ Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Architettura degli Elaboratori  
prova scritta di Assembly

ISTRUZIONI:

8. Scrivete subito il vostro nome, cognome e numero di matricola su questo foglio.
9. Aprite un editor a vostra scelta, scrivete una prima linea di commento con le stesse informazioni formattate nel seguente modo: *#matricola nome cognome*  
poi salvate il file chiamandolo: *cognomen.s* (con il vostro cognome e iniziale del nome!).
10. Tutti gli esercizi vanno scritti su questo file.
11. Per consegnare, bisogna risolvere i primi due esercizi e verificare con `xspim` che la soluzione sia corretta con i dati dell'esempio.
12. L'etichetta **main** va messa davanti alla funzione di test dell'esercizio 4 se ci arrivate e funziona altrimenti dovete metterla davanti al primo test (esercizio 2).
13. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
14. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

**Esercizio 1 (3 punti)**

Scrivere una funzione **noDup** che prende come parametri un puntatore ad una stringa e un carattere *c*. La funzione elimina tutte le eventuali occorrenze di *c* dalla stringa ad eccezione della prima. Ad esempio, dati: "trentatre" e 't'; al termine la stringa conterrà: "trenare".

**Esercizio 2 (1 punto)**

Scrivere una funzione **noDupTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **noDup** ai dati e stampa la stringa.

**Esercizio 3 (3 punti)**

Scrivere una funzione **dupFilter** che prende come parametro un puntatore a stringa. La funzione restituisce la stringa senza caratteri che si ripetono. La funzione deve chiamare la **noDup**. Ad esempio, dato "trentatre"; la funzione restituisce la stringa: "trena".

**Esercizio 4 (1 punto)**

Scrivere una funzione **dupFilterTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **dupFilter** ai dati e stampa la stringa.

NB. Si ricorda che per stampare un intero lo si deve caricare in `$a0`, caricare 1 in `$v0` e chiamare `syscall`; invece per stampare una stringa la si deve caricare in `$a0`, caricare 4 in `$v0` e chiamare `syscall`.