

Nome: _____ Cognome: _____ Matr.: _____

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 (2 punti)

Scrivere una funzione **vocale** che prende come parametro un carattere *c*. La funzione restituisce 1 se *c* è una vocale (minuscola) altrimenti restituisce 0. Ad esempio, dato: "a"; la funzione ritorna 1.

Esercizio 2 (1 punto)

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

Esercizio 3 (4 punti)

Scrivere una funzione **filtraVocali** che prende 2 parametri: un puntatore ad una stringa e un puntatore ad un secondo array. La funzione copia la stringa nel secondo array ad eccezione delle vocali. L'array deve essere terminato con uno zero; si assume che il secondo array sia abbastanza grande. Ad esempio, data la stringa: "Scrivere una"; la funzione scrive nel secondo array: "Scrivr n".

Esercizio 4 (1 punto)

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

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`.

Nome: _____ Cognome: _____ Matr.: _____

Architettura degli Elaboratori
prova scritta di Assembly

ISTRUZIONI:

1. Scrivete subito il vostro nome, cognome e numero di matricola su questo foglio.
8. 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!).
2. Tutti gli esercizi vanno scritti su questo file.
3. Per consegnare, bisogna risolvere i primi due esercizi e verificare con `xspim` che la soluzione sia corretta con i dati dell'esempio.
4. 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).
5. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
6. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

Esercizio 1 (2,5 punti)

Scrivere una funzione **spaces** che prende come parametri un puntatore ad una stringa e un puntatore ad un array di interi ad 8bit. La funzione scrive nell'array gli indici degli spazi contenuti nella stringa. L'array deve essere terminato con il valore -1. Ad esempio, data la stringa: "uno due tre"; al termine l'array conterrà: 3,7,-1.

Esercizio 2 (1,5 punti)

Scrivere una funzione **spacesTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **spaces** ai dati e stampa l'array (senza terminatore).

Esercizio 3 (3 punti)

Scrivere una funzione **underscore** che prende come parametro un puntatore ad una stringa. La funzione chiama **spaces** passandogli la stringa e un proprio buffer (allocato nella sezione dati) e sostituisce tutti gli spazi con degli '_'. Ad esempio, data la stringa: "a bcd e" alla fine la stringa contiene: "a_bcd_e".

Esercizio 4 (1 punto)

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

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`.

Nome: _____ Cognome: _____ Matr.: _____

Architettura degli Elaboratori
prova scritta di Assembly

ISTRUZIONI:

1. 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!).
2. Tutti gli esercizi vanno scritti su questo file.
3. Per consegnare, bisogna risolvere i primi due esercizi e verificare con `xspim` che la soluzione sia corretta con i dati dell'esempio.
4. 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).
5. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
6. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

Esercizio 1 (2,5 punti)

Scrivere una funzione **lowerCase** che prende come parametro un puntatore ad una stringa. La funzione modifica la stringa sostituendo le lettere maiuscole con delle minuscole (basta aggiungere 32). Ad esempio, la stringa: "eRo MaIUScoLo" viene modificata in: "ero maiuscolo".

Esercizio 2 (1 punto)

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

Esercizio 3 (3,5 punti)

Scrivere una funzione **compare** che prende come parametri due puntatori a stringhe. La funzione rende minuscole le due stringhe richiamando **lowerCase** su entrambe poi le confronta e restituisce 1 se sono uguali e zero altrimenti. Ad esempio, dati: "eRo MaIUScoLo", "ero maiusCOLO"; la funzione restituisce 1.

Esercizio 4 (1 punto)

Scrivere una funzione **compareTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. La funzione chiama **compare** e stampa il risultato.

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`.

Nome: _____ Cognome: _____ Matr.: _____

Architettura degli Elaboratori
prova scritta di Assembly

ISTRUZIONI:

1. Scrivete subito il vostro nome, cognome e numero di matricola su questo foglio.
10. 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!).
2. Tutti gli esercizi vanno scritti su questo file.
3. Per consegnare, bisogna risolvere i primi due esercizi e verificare con *xspim* che la soluzione sia corretta con i dati dell'esempio.
4. 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).
5. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
6. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

Esercizio 1 (3 punti)

Scrivere una funzione **variazione** che prende come parametri un puntatore ad un array di interi 8 bit con segno, un puntatore ad un array da usare per l'output, e un intero indicante la dimensione del primo array. Per ogni coppia consecutiva di numeri del primo array, la funzione scrive nel secondo array -1,0 oppure 1 a seconda che il primo numero della coppia sia rispettivamente maggiore, uguale o minore del secondo. L'array di output deve essere terminato con un 2.

Ad esempio, dati: 1,4,1,5,6,2 e 6; al termine l'array di output conterrà: 1-111-12.

Esercizio 2 (1,5 punti)

Scrivere una funzione **variazioneTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **variazione** ai dati e stampa l'array (senza terminatore).

Esercizio 3 (3 punti)

Scrivere una funzione **stessoAndamento** che prende come parametri due puntatori ad array di interi 8 bit con segno di uguale lunghezza e un intero indicante la dimensione di entrambi. La funzione deve chiamare **variazione** su ciascun array e poi confrontare gli array di output risultanti restituendo 1 se sono uguali, 0 altrimenti. Gli array di output possono essere allocati nella sezione dati.

Ad esempio, dati: 1,4,1,5,6,2 e 2,7,4,5,9,8 la funzione restituisce 1.

Esercizio 4 (0,5 punti)

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

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`.