

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 (3 punti)

Scrivere una funzione **removeFirst** che prende come parametri un puntatore ad una stringa e un carattere *c*. La funzione, se esiste, cancella la prima occorrenza di *c* dalla stringa e restituisce 1, altrimenti restituisce 0. Ad esempio, dati: "barba" e 'a'; la funzione ritorna 1 e la stringa diventa: "brba".

Esercizio 2 (1 punto)

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

Esercizio 3 (3 punti)

Scrivere una funzione **removeWhile1** che prende come parametri due puntatori a stringhe. La funzione chiama **removeFirst** passandole la prima stringa e il primo carattere della seconda stringa. La chiamata viene ripetuta anche per i caratteri successivi finché la funzione restituisce 1 o finiscono i caratteri della seconda stringa. Ad esempio, dati "carnevale" e "aeiae"; la funzione riduce la prima stringa a: "crnvale".

Esercizio 4 (1 punto)

Scrivere una funzione **removeWhile1Test** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **removeWhile1** ai dati e stampa la prima 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`.

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 **zeroDup** che prende come parametri un array di 10 interi e un intero *i* (tutti a 32 bit con segno). La funzione scrive nell'array uno 0 al posto di ogni (eventuale) occorrenza dell'intero *i* ad eccezione della prima. Ad esempio, dato l'array: 1, 2, 3, 4, 5, 6, 4, 4, 3, 2 e l'intero 4; al termine l'array risulta così: 1,2,3,4,5,6,0,0,3,2.

Esercizio 2 (1,5 punti)

Scrivere una funzione **zeroDupTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **zeroDup** ai dati e stampa l'array.

Esercizio 3 (3 punti)

Scrivere una funzione **bag2set** che prende come parametri due puntatori ad array di 10 interi a 32 bit con segno. La funzione chiama **zeroDup** passandogli il primo array e uno alla volta tutti gli interi del secondo. Ad esempio, dati: 1, 2, 3, 4, 5, 6, 4, 4, 3, 2 e 3,1,4,5,0,0,0,0,0,0; alla fine il primo array contiene: 1, 2, 3, 4, 5, 6, 0, 0, 0, 2.

Esercizio 4 (1 punto)

Scrivere una funzione **bag2setTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **bag2set** ai dati e stampa il primo array.

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 **findSharp** che prende come parametri un puntatore ad una stringa e un puntatore ad una funzione. La funzione cerca la prima occorrenza del carattere '#' nella stringa e ne restituisce il puntatore, se non lo trova termina saltando alla funzione passata. Ad esempio, dati: "a##o#a" e un puntatore; la funzione ritorna un puntatore alla stringa: "##o#a".

Esercizio 2 (1,5 punti)

Scrivere una funzione **findSharpTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **findSharp** ai dati e stampa la stringa a partire dal carattere '#' oppure il messaggio "assente".

Esercizio 3 (3 punti)

Scrivere una funzione **fillForm** che prende come parametri due puntatori a stringhe. La funzione, facendo uso di **findSharp**, sostituisce tutti i caratteri '#' della prima stringa con i caratteri della seconda stringa. Si assuma che la seconda stringa contenga tutti i caratteri necessari. Ad esempio, dati: "a##o#a" e "ncr"; la funzione modifica la prima stringa come segue: "ancora".

Esercizio 4 (1 punto)

Scrivere una funzione **fillFormTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. La funzione chiama **fillForm** e stampa la prima 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:

7. 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!).
8. Tutti gli esercizi vanno scritti su questo file.
9. Per consegnare, bisogna risolvere i primi due esercizi e verificare con `xspim` che la soluzione sia corretta con i dati dell'esempio.
10. 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).
11. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
12. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

Esercizio 1 (2,5 punti)

Scrivere una funzione **contaInScala** che prende come parametro un array di interi a 16 bit senza segno terminato con uno 0. La funzione ritorna il numero di elementi consecutivi in scala ($n, n+1, n+2\dots$) che ci sono all'inizio dell'array; ritorna 0 se l'array è vuoto. Ad esempio, dato l'array: 3, 4, 5, 6, 4, 4, 3, 2, 0; la funzione ritorna 4.

Esercizio 2 (1,5 punti)

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

Esercizio 3 (3 punti)

Scrivere una funzione **arrayScale** che prende come parametri un array come in esercizio 1 e un array buffer. La funzione chiama **contaInScala** in modo da scrivere per ogni scala del primo array la sua lunghezza nell'array di output. L'array di output deve essere terminato con uno 0. Ad esempio, dati: 2, 4, 5, 6, 4, 4, 2, 3, 0; alla fine l'array di output contiene: 1, 3, 1, 1, 2.

Esercizio 4 (1 punto)

Scrivere una funzione **arrayScaleTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **arrayScale** ai dati e stampa l'array 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`.