

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 **prodij** che prende come parametri due puntatori ad array di interi a 32bit senza segno e due indici i e j . La funzione restituisce la componente i del primo array moltiplicata per la componente j del secondo. Si assume che il risultato sia ancora a 32 bit. Ad esempio, dati gli array: 1,3,5,7,9,11,13,15,17,19 e 2,4,6,8,10,12,14,16,18,20 e gli indici 1 e 5; la funzione restituisce il valore: 36 (3x12).

Esercizio 2 (1.5 punti)

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

Esercizio 3 (3.5 punti)

Scrivere una funzione **mono** che prende come parametri due puntatori ad array di interi a 32bit senza segno e la lunghezza comune degli array n . La funzione somma i risultati di **prodij** calcolata sulle n coppie di indici i,j con i,j che vanno rispettivamente da $n-1,0$ a $0,n-1$. Ad esempio, dati gli array dell'esercizio 1 con lunghezza 10; la funzione restituisce 770 ($19 \times 2 + 17 \times 4 + 15 \times 6 + \dots + 1 \times 20$).

Esercizio 4 (1 punto)

Scrivere una funzione **monoTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **mono** 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`.

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 punti)

Scrivere una funzione **prodi** che prende come parametri due puntatori ad array di interi a 32bit senza segno e un indice i . La funzione restituisce la componente i del primo array moltiplicata per la componente i del secondo. Si assume che il risultato sia ancora a 32 bit. Ad esempio, dati gli array: 1,3,5,7,9,11,13,15,17,19 e 2,4,6,8,10,12,14,16,18,20 e l'indice 5; la funzione restituisce il valore: 132 (11x12).

Esercizio 2 (1.5 punti)

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

Esercizio 3 (3.5 punti)

Scrivere una funzione **vect** che prende come parametri due puntatori ad array di interi a 32bit senza segno, un puntatore ad un array per i risultati e la lunghezza comune degli array n . La funzione scrive in ogni componente k dell'array dei risultati il valore di **prodi** calcolata sull'indice k . Ad esempio, dati gli array dell'esercizio 1 con lunghezza 10; la funzione restituisce 2,12,30,56,90,132,182,240,306,380 (1x2,3x4,5x6,...,19x20).

Esercizio 4 (1 punto)

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

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`.