

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 e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 **sum3Int** che prende come parametri un puntatore ad un array di interi a 16 bit con segno e un intero a 32 bit senza segno n . La funzione restituisce la somma dei primi n interi dell'array. Si assuma che l'array contenga almeno gli interi richiesti. Ad esempio, dati: un puntatore a 1,32,12 e 3; la funzione ritorna: 45.

Esercizio 2 (1 punto)

Scrivere una funzione **sum3IntTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **sum3Int** ai dati e stampa il risultato.

Esercizio 3 (3 punti)

Scrivere una funzione **sum3Copy** che prende come parametri un puntatore ad un array di interi a 16 bit con segno e la lunghezza dell'array (multiplo di 3). La funzione chiama per ogni tripletta **sum3Int** e stampa il risultato a 32 bit. Ad esempio, dato l'array: 1, 32, 12, 3, 6, 1, 0, 3, 3, 3,4,2 e 12; la funzione stampa: 45 10 6 9.

Esercizio 4 (2 punti)

Scrivere una funzione **sum3CopyTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **sum3Copy** ai dati.

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.
2. Aprite un editor a vostra scelta, scrivete una prima linea di commento con le stesse informazioni e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 **strCopy** 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 compresi. Il buffer deve essere zero terminato. Si assuma che gli indici sono interni alla stringa. Ad esempio, data la stringa: "uno %due% tre", 5, 7; la funzione scrive nel buffer: "due".

Esercizio 2 (2 punti)

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

Esercizio 3 (3 punti)

Scrivere una funzione **extractStr** 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 **strCopy** 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: "uno %due% tre" e il caratteri '%'; alla fine il buffer contiene: "due".

Esercizio 4 (1 punto)

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

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 e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 **binOp** che prende come parametri un codice (a 32 bit) e due interi a 32 bit con segno. La funzione restituisce la somma dei due interi se il codice è 0 mentre se è 1 restituisce la differenza. Ad esempio, dati: 0, 5, 7; la funzione ritorna 12.

Esercizio 2 (1,5 punti)

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

Esercizio 3 (3,5 punti)

Scrivere una funzione **binExp** che prende come parametro un puntatore ad un array di interi a 32 bit con segno. L'array alterna interi e operazioni (vedi esercizio 1) come segue: intero op intero op intero, ecc. ed è terminato con un due. La funzione chiama **binOp** sui primi due interi poi sul risultato ed il terzo intero e così via; alla fine restituisce il risultato finale. Ad esempio, dato l'array: 5, 0, 7, 0, 3, 1, 5, 2; la funzione restituisce 10.

Esercizio 4 (1 punto)

Scrivere una funzione **binExpTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. La funzione chiama **binExp** 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.
2. Aprite un editor a vostra scelta, scrivete una prima linea di commento con le stesse informazioni e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 **insertAt** che prende tre parametri: un puntatore ad un buffer stringa, un puntatore a stringa e un indice a 32 bit. La funzione inserisce la stringa nel buffer a partire dall'indice dato. Ad esempio, dati: "scrivere funzione", "una " e 9; la funzione restituisce: "scrivere una ione".

Esercizio 2 (2 punti)

Scrivere una funzione **insertAtTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 1. La funzione chiama **insertAt** e stampa il buffer.

Esercizio 3 (3 punti)

Scrivere una funzione **makeSpace** che prende tre parametri: un puntatore ad un buffer stringa, un indice a 32bit e una spostamento a 32bit senza segno. La funzione copia avanti di *spostamento* byte i caratteri della stringa a partire dall'indice dato. Ad esempio, dato un buffer contenente: "scrivere funzione" e 8, 4; la funzione restituisce: "scrivere fun funzione".

Esercizio 4 (1 punto)

Scrivere una funzione **makeSpaceTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 3. Il test applica la funzione **makeSpace** 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:

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 e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 (2 punti)

Scrivere una funzione **seqGen** che prende come parametri due interi a 8 bit senza segno. La funzione stampa il primo intero e i suoi multipli per un totale di risultati pari al secondo intero. Ad esempio, dati: 3, 5; la funzione stampa 3, 6, 9, 12, 15.

Esercizio 2 (2 punti)

Scrivere una funzione **seqGenTest** senza parametri e con una sezione dati contenente i dati dell'esempio dell'esercizio 1. Il test applica la funzione **seqGen** ai dati.

Esercizio 3 (3 punti)

Scrivere una funzione **seqArray** che prende come parametro un puntatore ad un array di interi a 8 bit senza segno e un intero senza segno. La funzione chiama **seqGen** su tutti gli elementi dell'array (0 terminato) passandogli l'intero come secondo parametro. Ad esempio, dato l'array: 1, 2, 3, 0 e 4; la funzione stampa: 1,2,3,4,2,4,6,8,3,6,9,12.

Esercizio 4 (1 punto)

Scrivere una funzione **seqArrayTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **seqArray** ai dati.

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:

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 e salvate il file chiamandolo: *cognome.s* (con il vostro cognome!).
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 (2 punti)

Scrivere una funzione **delimIdx** che prende due parametri: un puntatore ad una stringa e un carattere. La funzione restituisce in *\$v0* e *\$v1* rispettivamente gli indici delle prime due occorrenze del carattere nella stringa. Si assuma che esistono. Ad esempio, data la stringa: "uno due tre" e ' '; la funzione restituisce: 3, 7.

Esercizio 2 (2 punti)

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

Esercizio 3 (3 punti)

Scrivere una funzione **extractStr** che prende due parametri: un puntatore ad una stringa e un carattere. La funzione stampa la prima sottostringa delimitata dal carattere nella stringa data. La funzione deve richiamare **delimIdx**. Ad esempio, dati: "uno due tre", e " "; la funzione stampa: "due".

Esercizio 4 (1 punto)

Scrivere una funzione **extractStrTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **extractStr** ai dati.

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*.