

Nome: _____ Cognome: _____ Matr.: _____
Fila: _____ Posto: _____

Parte C

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*
3. Per consegnare, bisogna risolvere almeno uno dei primi due esercizi più il terzo e verificare con `xspim` che la soluzione sia corretta (almeno) con i dati dell'esempio dato.
4. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
5. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

C/Esercizio 1 (2 punti)

Scrivere una funzione **reorder** che prende come parametro un puntatore ad un array di 4 interi a 16 bit senza segno. La funzione modifica l'array scambiando la posizione dei 4 interi come segue: 1,2,3,4 in 3,4,1,2. Ad esempio, dato l'array: 25, 2, 5, 13; la funzione ritorna: 5, 13, 25, 2.

C/Esercizio 2 (3 punti)

Scrivere una funzione **reorderAll** che prende come parametro un puntatore ad un array di interi a 16 bit senza segno. L'array è terminato con uno zero ed ha una lunghezza multiplo di 4 interi. La funzione modifica l'array riordinando tutti i gruppi di 4 interi usando la funzione **reorder**. Ad esempio, dato l'array: 1, 2, 3, 4, 5, 6, 7, 8, 0; la funzione ritorna: 3, 4, 1, 2, 7, 8, 5, 6, 0.

C/Esercizio 3 (3 punti)

Scrivere una funzione **reorderTest** senza parametri; con una sezione dati contenente l'array dell'esempio precedente. La funzione applica **reorderAll** ai dati e stampa l'intero array.

NB. Se fate solo l'esercizio 1, dovete adattare questa funzione in modo da verificare il suo funzionamento.

Si ricorda che per stampare un intero lo si deve caricare in `$a0`, caricare 1 in `$v0` e chiamare `syscall`.

Si ricorda inoltre di assegnare anche l'etichetta **main** a questa funzione per poterla testare con `xspim`.

Nome: _____ Cognome: _____ Matr.: _____
Fila: _____ Posto: _____

Parte C

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*
3. Per consegnare, bisogna risolvere almeno uno dei primi due esercizi più il terzo e verificare con `xspim` che la soluzione sia corretta (almeno) con i dati dell'esempio dato.
4. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
5. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

C/Esercizio 1 (2.5 punti)

Scrivere una funzione **contaUguali** che prende come parametro un puntatore ad un array di interi a 16 bit senza segno. L'array è terminato con uno zero. La funzione restituisce quante volte il primo numero è ripetuto consecutivamente all'inizio dell'array. Se l'array è vuoto deve restituire 0. Ad esempio, dato l'array: 25, 25, 25, 13, 81, 14, 0; la funzione ritorna 3.

C/Esercizio 2 (3.5 punti)

Scrivere una funzione **arrayVolte** che prende 2 parametri: un puntatore ad un array come da esercizio 1 e un puntatore ad un secondo array a 8 bit. La funzione scrive nel secondo array per ogni sequenza di numeri uguali il numero di volte che si ripetono (dato da **contaUguali**). L'array deve essere terminato con uno zero; si assume che il secondo array sia abbastanza grande. Ad esempio, dato l'array: 25, 25, 25, 13, 81, 81, 4, 0; la funzione scrive nel secondo array: 3, 1, 2, 1, 0.

C/Esercizio 3 (2 punti)

Scrivere una funzione **volteTest** senza parametri; con una sezione dati contenente i due array dell'esempio precedente. La funzione passa ad **arrayVolte** i due array e stampa l'array risultante.

NB. Se fate solo l'esercizio 1, dovete adattare questa funzione in modo da verificare il suo funzionamento.

Si ricorda che per stampare un intero lo si deve caricare in `$a0`, caricare 1 in `$v0` e chiamare `syscall`.

Si ricorda inoltre di assegnare anche l'etichetta **main** a questa funzione per poterla testare con `xspim`.

Nome: _____ Cognome: _____ Matr.: _____
Fila: _____ Posto: _____

Parte C

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*
3. Per consegnare, bisogna risolvere almeno uno dei primi due esercizi più il terzo e verificare con `xspim` che la soluzione sia corretta (almeno) con i dati dell'esempio dato.
4. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
5. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

C/Esercizio 1 (2.5 punti)

Scrivere una funzione **trimStart** che prende come parametro un puntatore ad una stringa. La funzione restituisce un puntatore ad una stringa senza spazi all'inizio. Ad esempio, data la stringa: " esempio di stringa "; la funzione ritorna la stringa "esempio di stringa".

C/Esercizio 2 (3.5 punti)

Scrivere una funzione **trim** che prende come parametro un puntatore ad una stringa. La funzione restituisce un puntatore ad una stringa senza spazi ne all'inizio ne alla fine. Per togliere gli spazi iniziali deve chiamare **trimStart**. Ad esempio, data la stringa: " esempio di stringa "; la funzione ritorna "esempio di stringa".

C/Esercizio 3 (2 punti)

Scrivere una funzione **trimTest** senza parametri; con una sezione dati contenente la stringa dell'esercizio

1. La funzione applica le funzione **trim** alla stringa e stampa la stringa risultante.

NB. Se fate solo l'esercizio 1, dovete adattare questa funzione in modo da verificare il suo funzionamento.

Si ricorda che per stampare una stringa la si deve caricare in `$a0`, caricare 4 in `$v0` e chiamare `syscall`.

Si ricorda inoltre di assegnare anche l'etichetta **main** a questa funzione per poterla testare con `xspim`.

Nome: _____ Cognome: _____ Matr.: _____
Fila: _____ Posto: _____

Parte C

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*
3. Per consegnare, bisogna risolvere almeno uno dei primi due esercizi più il terzo e verificare con `xspim` che la soluzione sia corretta (almeno) con i dati dell'esempio dato.
4. Chi consegna e ha rispettato quanto scritto sopra prende almeno 3 punti (la sufficienza).
5. In ogni caso, prima di uscire dovete restituirmi questo foglio e dirmi se consegnate oppure no.

C/Esercizio 1 (3 punti)

Scrivere una funzione `commonPrefixSize` che prende come parametri due puntatori a stringa. La funzione restituisce la lunghezza del prefisso comune. Ad esempio, date le stringhe: "prefisso uno" e "prefisso due"; la funzione restituisce: 9.

C/Esercizio 2 (3 punti)

Scrivere una funzione `commonPrefix` che prende come parametri due puntatori a stringa. La funzione restituisce il prefisso comune. La funzione deve richiamare `commonPrefixSize` e può costruire il risultato modificando uno dei parametri. Ad esempio, date le stringhe: "prefisso_uno" e "prefisso_due"; la funzione restituisce: "prefisso_".

C/Esercizio 3 (2 punti)

Scrivere una funzione `commonPrefixTest` senza parametri; con una sezione dati contenente le stringhe dell'esercizio 1. La funzione chiama `commonPrefix` e stampa la stringa risultante.

NB. Se fate solo l'esercizio 1, dovete adattare questa funzione in modo da verificare il suo funzionamento.

Si ricorda che per stampare un intero lo si deve caricare in `$a0`, caricare 1 in `$v0` e chiamare `syscall`.

Si ricorda che per stampare una stringa la si deve caricare in `$a0`, caricare 4 in `$v0` e chiamare `syscall`.

Si ricorda inoltre di assegnare anche l'etichetta `main` a questa funzione per poterla testare con `xspim`.