

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_ Matr.: \_\_\_\_\_

## 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* (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 **result** che prende come parametri due interi ad 8 bit senza segno. La funzione restituisce "1" se il primo intero è maggiore del secondo, "x" se sono uguali, altrimenti "2". Ad esempio, dati: 1 e 3; la funzione ritorna: 2.

#### Esercizio 2 (2 punti)

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

#### Esercizio 3 (3 punti)

Scrivere una funzione **resultAll** che prende come parametro un puntatore a un array di 6 coppie di interi a 8 bit senza segno. La funzione chiama per ogni coppia **result** e stampa il risultato. Ad esempio, dato l'array: 1, 2, 3, 4, 1, 1, 0, 3, 3, 2, 2, 2; la funzione stampa: 22x21x.

#### Esercizio 4 (1 punto)

Scrivere una funzione **resultAllTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **resultAll** 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.: \_\_\_\_\_

## 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* (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 **replace** con tre parametri: un puntatore ad una stringa e due caratteri. La funzione modifica la stringa sostituendo tutte le occorrenze del primo carattere con il secondo. Ad esempio, data la stringa: "stringa da modificare" e i caratteri " " e "\_"; la funzione modifica la stringa come segue: "stringa\_da\_modificare".

#### Esercizio 2 (2 punti)

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

#### Esercizio 3 (3 punti)

Scrivere una funzione **words** che prende come parametro un puntatore ad una stringa. La funzione divide la stringa in parole e restituisce la prima parola. Per dividere in parole deve sostituire gli spazi con degli 0 chiamando **replace**. Ad esempio, data la stringa: "uno due tre"; la funzione ritorna "uno".

#### Esercizio 4 (1 punto)

Scrivere una funzione **wordsTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. Il test applica la funzione **words** 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.: \_\_\_\_\_

## 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* (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 **orderedPrefix** che prende come parametro un puntatore ad un array di interi a 32 bit senza segno. L'array è terminato con uno zero. La funzione restituisce il numero di interi che sono in ordine strettamente crescente all'inizio dell'array. Ad esempio, dato l'array: 2, 5, 7, 7, 6, 25, 0; la funzione ritorna 3.

#### Esercizio 2 (2 punti)

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

#### Esercizio 3 (3 punti)

Scrivere una funzione **isOrdered** che prende come parametro un puntatore ad un array di interi a 32 bit senza segno. L'array è terminato con uno zero. La funzione restituisce 1 se l'array è tutto ordinato altrimenti 0. La funzione deve chiamare **orderedPrefix**. Ad esempio, dato l'array: 2, 5, 15, 25, 30, 0; la funzione restituisce 1.

#### Esercizio 4 (1 punto)

Scrivere una funzione **isOrderedTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. La funzione chiama **isOrdered** 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.: \_\_\_\_\_

## 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* (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 **checkLess** che prende due parametri: un puntatore a stringa e un carattere. La funzione restituisce 1 se tutti i caratteri della stringa sono minori o uguali al carattere passato; altrimenti restituisce 0. Ad esempio, data la stringa: "abcecba" e "d"; la funzione restituisce: 0.

#### Esercizio 2 (2 punti)

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

#### Esercizio 3 (3 punti)

Scrivere una funzione **checkRange** che prende tre parametri: un puntatore a stringa e due caratteri. La funzione restituisce 1 se tutti i caratteri della stringa sono compresi tra i due caratteri passati altrimenti restituisce 0. La funzione deve richiamare **checkLess**. Ad esempio, dati: "qwerty", "a" e "z"; la funzione restituisce: 1.

#### Esercizio 4 (1 punto)

Scrivere una funzione **checkRangeTest** senza parametri; con una sezione dati contenente i dati dell'esercizio 3. La funzione chiama **checkRange** 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`.