

Esercizi di programmazione Assembler

Gli esercizi di programmazione assembler riportati di seguito comprendono tutti gli esempi fatti a lezione e sono da eseguirsi nell'ordine dato.

Programmi iniziali

ok.asm Stampa la scritta OK.

pari.asm Preleva dalla memoria un intero a 16 bit (**num**) e verifica che sia divisibile per due. Come risultato scrive 1 nella cella **risu** se il numero è pari, altrimenti scrive 0. Quindi stampa sul video il contenuto (0 o 1) della cella **risu**.

ESEMPIO: se **num = 10**, allora pone **risu = 1** e stampa **1**.
se **num = 5**, allora pone **risu = 0** e stampa **0**.

bitk.asm Preleva dalla memoria due numeri interi a 16 bit (**num** e **kappa**) e scrive 1 nella cella **risu** se il bit **kappa** di **num** è uguale a 1, altrimenti scrive 0. Quindi stampa sul video il contenuto (0 o 1) della cella **risu**.

ESEMPIO: se **num = 000FH** e **kappa = 3**, allora pone **risu = 1** e stampa 1,
se **num = 000FH** e **kappa = 4**, allora pone **risu = 0** e stampa 0.

porte.asm Preleva dalla memoria due numeri interi a 16 bit (**num** e **kappa**) e scrive 1 nella cella **risu** se i primi **kappa** bit di **num** sono tutti uguali a 1, altrimenti scrive 0. Quindi stampa sul video il contenuto (0 o 1) della cella **risu**.

ESEMPIO: se **num = 000FH** e **kappa = 4**, allora pone **risu = 1** e stampa 1,
se **num = 0FF7H** e **kappa = 4**, allora pone **risu = 0** e stampa 0.

vett10.asm Definisce un vettore di 10 interi da 16 bit inizializzati con numeri da 0 a 9. Quindi stampa il contenuto dei dieci elementi del vettore.

STAMPA: 0123456789

vett16.asm Definisce un vettore di 16 interi da 16 bit inizializzati con numeri da 0 a 15. Quindi stampa il contenuto esadecimale dei sedici elementi del vettore.

STAMPA: 0123456789ABCDEF

stringa.asm Definisce una stringa di byte inizializzata con la parola 'buongiorno'. Quindi stampa la stringa sul video.

STAMPA: buongiorno

Procedure con parametri nei registri

Le procedure seguenti vanno definite nello stesso file che contiene il programma principale, posizionandole subito dopo la chiamata INT 21H del ritorno a DOS.

ok2.asm Tale programma richiama due volte una procedura di nome **okay** che stampa la scritta OK. La procedura **okay** non riceve parametri.

STAMPA: OKOK

vett2.asm Il programma definisce un vettore di 10 interi da 16 bit inizializzati con numeri da 0 a 9 e richiama una procedura di nome **printv** che stampa gli elementi del vettore. La procedura **printv** si aspetta l'indirizzo del vettore in BX e la dimensione in CX.

STAMPA: 0123456789

string2.asm Definire due stringhe di byte **s1** ed **s2** inizializzate con le parole '**primo**' e '**secondo**'. Quindi scrivere una procedura di nome **prints** che stampi una stringa il cui indirizzo viene passato nel registro BX. Infine scrivere un programma che in sequenza richiami la procedura **prints** con la stringa **s1**, stampi uno spazio, e poi richiami una seconda volta la procedura **prints** con la stringa **s2**.

STAMPA: primo secondo

tasto.asm Definire due stringhe **s1** ed **s2** inizializzate con le frasi '**Premi un tasto**' e '**Hai premuto il tasto**'. Quindi scrivere un programma che richiami la procedura **prints** con la stringa **s1** e poi si blocchi in attesa della pressione di un tasto sulla tastiera. Quando il tasto viene premuto, il programma deve stampare un ritorno a capo, richiamare una seconda volta la procedura **prints** con la stringa **s2** e poi stampare il carattere corrispondente al tasto premuto.

STAMPA: Premi un tasto <si preme F>
Hai premuto il tasto F

Procedure con parametri nello stack

Le seguenti procedure devono essere definite in un file diverso da quello del programma principale, chiamato **mylib.asm**, in modo da costituire un file di libreria da linkare con il programma principale, il quale ha il compito di richiamarle con parametri diversi.

space	Stampa uno spazio. Non riceve parametri.
newline	Stampa un ritorno a capo. Non riceve parametri.
clear	Cancella lo schermo. Non riceve parametri.
stampac	Riceve nello stack la codifica ASCII di un carattere e lo stampa.
stampas	Riceve nello stack l'indirizzo di una stringa e la stampa.
stampab	Riceve nello stack un intero assoluto a 16 bit e lo stampa in binario, sempre su 16 cifre.
stampax	Riceve nello stack un intero assoluto a 16 bit e lo stampa in esadecimale, sempre su 4 cifre.
stampad	Riceve nello stack un intero relativo a 16 bit e lo stampa in decimale. La procedura deve stampare il segno meno se il numero è negativo e il carattere '0' se il numero è uguale a zero.
lungs	Riceve nello stack l'indirizzo di una stringa e restituisce in AX la sua lunghezza, ossia il numero di caratteri che la compongono (escluso lo zero).
maius	Riceve nello stack l'indirizzo di una stringa e la stampa in maiuscolo.
inverti	Riceve nello stack l'indirizzo di una stringa e la stampa al contrario, dall'ultimo carattere al primo.
vocal	Riceve nello stack l'indirizzo di una stringa e restituisce in AX il numero di vocali in essa contenute.
uni	Riceve nello stack un intero assoluto a 16 bit e restituisce in AX il numero di uni in esso contenuti.