

a.a. 2014/15 - Architettura degli Elaboratori  
ESEMPI DI DOMANDE D'ESAME, PER VALUTARE LA PROPRIA PREPARAZIONE.

-----

Cos'è l'interprete dei comandi e a che livello dell'architettura degli elaboratori si trova?

Cos'è il bootstrap?

Cos'è il BIOS?

E' possibile per un utente creare delle proprie librerie, oppure l'utente può usare solo delle librerie "di sistema"?

Cos'è la memoria ROM?

Cosa sono i BUS in un computer?

Nel contesto dell'Architettura degli Elaboratori, cos'è la CACHE per la memoria?

Cosa si intende con Endianess? Fare un esempio che spieghi le differenze tra Big Endian e Little Endian.

A cosa serve la funzione di libreria del linguaggio C denominata **htonl ()** ?

Il risultato restituito dalla chiamata alla funzione di libreria `htons( 30000 )`; cambia al variare della Endianess del processore su cui viene eseguita?

Il risultato restituito dalla chiamata alla funzione di libreria `ntohs( 30000 )`; cambia al variare della Endianess del processore su cui viene eseguita?

Supponiamo di voler rappresentare, nella memoria di un computer, una variabile intera K che può assumere valori tra 0 e 65535. Supponiamo che questa variabile sia dichiarata e utilizzata all'interno di un programma scritto in linguaggio ANSI C. Il programma può assegnare dei valori a questa variabile K (rimanendo nell'intervallo di valori previsto) specificandoli mediante diverse notazioni che usano diverse basi di rappresentazione. Ad esempio  $K=19$ ;  $K=0x19$ ;  $K=b10001$ ;  
Si consideri la dimensione in byte della variabile K: questa dimensione cambia al variare della base di rappresentazione utilizzata per specificare i valori da assegnare?  
Motivare la risposta.

Scrivere l'espressione che permette di esprimere il valore `0xC4` esplicitandolo secondo la base decimale. Non occorre svolgere i calcoli, basta scrivere correttamente l'espressione.

Una struttura dati S occupa in memoria il blocco di byte aventi indirizzo compreso tra 1000 e 1007 (compresi gli estremi). Qual è l'indirizzo di quella struttura dati? Qual è la dimensione in byte di quella struttura dati?

Elencare un insieme di valori interi che possono essere rappresentati con 3 bit.

1) Quanti bit contiene un Byte?

2) Quanti diversi numeri sono rappresentabili con un intero a 16 bit? Si può rispondere anche con una formula (senza incognite) senza svolgere i calcoli.

3) Quanti diversi numeri sono rappresentabili con un intero a 4 bytes? Si può rispondere anche con una formula (senza incognite) senza svolgere i calcoli.

4) Dato un intero senza segno a 16 bit, di valore 0xAABB, collocato all'indirizzo 0x1000 di memoria in un computer con ordinamento dei bytes di tipo Big Endian, quali sono i valori in memoria dei singoli bytes in cui quell'intero è collocato? Si usi la rappresentazione in memoria qui di seguito proposta, indicando i valori dei soli bytes che compongono l'intero. Esprimere i valori dei bytes secondo una base numerica a vostra scelta.

Address	Value
0x1004	
0x1003	
0x1002	
0x1001	
0x1000	
0x0FFF	
0x0FFE	
0x0FFD	

5) Dato un intero senza segno a 32 bit, collocato all'indirizzo 0x1000 di memoria in un computer con ordinamento dei bytes di tipo Little Endian, sapendo che i valori in memoria sono quelli rappresentati nella seguente figura, qual è il valore di quell'intero? Esprimere l'intero secondo una base numerica a vostra scelta.

Address	Value
0x1004	0xFF
0x1003	0xEE
0x1002	0xDD
0x1001	0xCC
0x1000	0xBB
0x0FFF	0xAA
0x0FFE	0x99
0x0FFD	0x88

6) Esprimere in base 10, il valore dell'intero a 16 bit 0xA00D. Si può rispondere anche con una formula (senza incognite) senza svolgere i calcoli.

7) Qual è la base numerica in cui viene rappresentato, in linguaggio C, questo numero? 0b1010

8) Qual è la base numerica in cui viene rappresentato, in linguaggio C, questo numero? 0x1010

Durante l'esecuzione di un programma, dove sono contenute le istruzioni che compongono il programma?

Cos'è il ciclo di data path ?

Cos'è il ciclo di Neumann ?

Fornire un esempio che spieghi perché l'esecuzione del ciclo di Neumann comporta l'esecuzione di molti cicli di data path.

In un processore con architettura Big Endian, sia data in memoria RAM una variabile intera, senza segno, a 16 bit. Tale variabile si trova all'indirizzo 1000. Il byte di indirizzo 1000 contiene il valore 2. Il byte di indirizzo 1001 contiene il valore 1. Qual è il valore della variabile intera? Esprimere il valore secondo una base di rappresentazione a vostra scelta, eventualmente anche mediante un'espressione senza svolgere i calcoli.

La dimensione in memoria di una variabile di tipo vettore di caratteri ( ad esempio `char vet[10]` ), secondo il linguaggio ANSI C, dipende o non dipende dalla dimensione della word tipica di quel processore del processore? Perché?

La dimensione in memoria di una variabile di tipo `int32_t` , secondo il linguaggio ANSI C, dipende o non dipende dalla endianess del processore? Perché?

Che cosa si intende quando si parla dice, informalmente, che un processore “ è a 32 bit ” oppure che “ è a 64 bit ” ?

Con riferimento all'Architettura degli Elaboratori, indicare che cos'è e a che cosa serve il BUS Dati.

Con riferimento all'Architettura degli Elaboratori, indicare che cos'è e a che cosa serve il BUS degli Indirizzi.

Con riferimento all'Architettura degli Elaboratori, indicare che cos'è e a che cosa serve il BUS di controllo.

Cosa si intende quando si parla di “WORD” o “PAROLA” di un processore ?

Che cosa si intende con Ampiezza del Bus degli Indirizzi?

Indicare qual è la relazione tra l'ampiezza del BUS degli indirizzi e la quantità di memoria indirizzabile dal processore che utilizza quel BUS.

La dimensione in memoria di una stessa struttura dati (struct), composta da diversi campi, è sempre la stessa in tutti i processori? Motivare la risposta.

Descrivere come viene rappresentata, in un file di testo, la fine di una riga, cioè l'andata a capo riga. Indicare in particolare se tale rappresentazione dipende dal sistema operativo.

Quanti bit occupa un carattere codificato secondo la codifica ASCII? E quanti caratteri diversi vengono rappresentati con la codifica ASCII?

Quanti caratteri vengono rappresentati con una codifica ad 8 bit, ad esempio la codifica iso-8859-1 (anche detta Latin-1)? Che relazione esiste tra la codifica ASCII e la citata codifica ad 8 bit iso-8859-1 (anche detta Latin-1)?

Quanti byte sono necessari per rappresentare in memoria il carattere 'a' secondo la codifica ASCII?  
Quanti byte sono necessari per rappresentare in memoria il carattere 'a' secondo la codifica iso-8859-1 (anche detta Latin-1)?

Quanti byte sono necessari per rappresentare in memoria il carattere 'a' secondo la codifica multibyte UTF-8 ?

I caratteri rappresentati dalla codifica ASCII sono rappresentati nello stesso modo anche nella codifica multibyte UTF-8?

I caratteri rappresentati dalla codifica a 8 bit iso-8859-1 (anche detta Latin-1) sono rappresentati nello stesso modo anche nella codifica multibyte UTF-8?

Esiste qualche codifica di caratteri che dipende dall'endianess del processore? In caso affermativo fornire un esempio.

Cos'è il BOM (Byte Order Mark) e in che contesto viene utilizzato?

Con riguardo all'interprete dei comandi (o anche shell di comandi), qual è la differenza tra un comando ed un eseguibile?

Un programmatore che utilizza il linguaggio C chiama direttamente le system call messe a disposizione dal sistema operativo? In caso contrario, come fa un programmatore ad invocare le system call mediante il linguaggio ANSI C?

Le librerie di sistema per il linguaggio ANSI C cosa contengono, essenzialmente?

La comune e diffusissima interfaccia di programmazione per il linguaggio ANSI C messa a disposizione dalle POSIX API, varia passando da un sistema operativo ad un altro ( ovviamente per sistemi operativi che supportano lo standard POSIX ) ? Motivare la risposta.

L'implementazione della comune e diffusissima interfaccia di programmazione per il linguaggio ANSI C messa a disposizione dalle POSIX API, varia passando da un sistema operativo ad un altro (ovviamente per sistemi operati che supportano lo standard POSIX) ? Motivare la risposta.

Cos'è una partizione di un harddisk?

Cosa si intende quando si dice in gergo tecnico (e un po' imprecisamente) che "una partizione è montata in un punto di un file system ?

Con riferimento alla shell di comandi bash, spiegare cosa si intende con i termini "comando", "file eseguibile binario" e "script".

Quando un utente ordina alla shell di comandi bash di eseguire un certo file eseguibile, digitandone il nome, ma senza specificare il percorso completo (assoluto o relativo) per raggiungere quel file, in che modo la shell trova l'eseguibile da eseguire?

In una shell di comandi bash, spiegare la differenza tra "variabili locali" e "variabili di ambiente".

Scrivere in modo opportuno gli argomenti da passare al comando **echo**, in modo che tale comando stampi sulla console la seguente sequenza di caratteri:

```
casa ; cd /bin
```

In una shell di comandi bash, scrivere un semplice script che concatena in una stessa variabile , denominata STRINGA, tutti gli argomenti passati a riga di comando allo script.

Per esempio, se lo script viene invocato passando come argomenti i seguenti

aaa "bb cccc" dd fffff

allora, alla fine dello script, la variabile STRINGA deve contenere aaabb ccccdfffff

In una shell di comandi bash, che cos'è la "directory corrente" ?

In una shell di comandi bash, a cosa serve il comando cat ?

In una shell di comandi bash, a cosa serve la variabile \$? ?

In una shell di comandi bash, scrivere una semplice riga di comandi che mette in una variabile STRINGA l'output prodotto dal comando ls -alh .

In una shell di comandi bash, scrivere una sequenza di comandi in modo da ottenere questi risultati:

- 1) cambiare la directory corrente spostandosi nella directory /pippo
- 2) Solo se lo spostamento ha avuto buon fine, visualizzare a video l'elenco dei file contenuti nella directory corrente.

In una shell di comandi bash, scrivere una sequenza di comandi in modo da ottenere questi risultati:

- 1) copiare il contenuto del file /home/vic/testo.txt nel file /home/tutto.txt
- 2) Solo se la copia NON è andata a buon fine, visualizzare a video il seguente messaggio: mannaggia

In una shell di comandi bash, scrivere una sequenza di comandi in modo da ottenere questi risultati:

- 1) eseguire l'eseguibile myexe.exe che si trova nella directory /home/vic/bin
- 2) Controllare il risultato restituito dall'eseguibile. Se il risultato vale 3 allora visualizzare a video il contenuto della directory corrente.

In una shell di comandi bash, se scrivo uno script, in che modo posso specificare il risultato numerico intero che quello script deve restituire?

In una shell di comandi bash, scrivere una sequenza di comandi in modo da ottenere questi risultati: viene eseguito l'eseguibile /bin/esegui.exe passandogli come standard input il contenuto del file di testo /home/vic/testo.txt. L'output, su standard output, prodotto dell'eseguibile viene a sua volta passato come standard input ad un altro eseguibile /sbin/usa.exe

-----

**7605)** Un programma è formato da più files sorgenti scritti in linguaggio ANSI C, qui di seguito elencati.

Il file costanti.h non include altri file.

Il file strutture.h include il file costanti.h

Il file strutture.c include il file strutture.h

Il file main.c include il file costanti.h ed il file funzioni.h

Il file funzioni.c non include nessun file.

Scrivere un Makefile che utilizzi il compilatore gcc per generare il file eseguibile progetto.exe partendo dai files citati e che sia in grado di limitare la necessità di ricompilare tutti i files quando viene modificato solo uno di questi.

**7606)** Un programma è formato da più files sorgenti scritti in linguaggio ANSI C, qui di seguito elencati.

Il file `costanti.h` non include altri file.

Il file `strutture.h` include il file `costanti.h`

Il file `strutture.c` include il file `strutture.h`

Il file `main.c` include il file `costanti.h` ed il file `funzioni.h`

Il file `funzioni.c` non include nessun file.

Scrivere una riga di comando che faccia eseguire il compilatore `gcc` per effettuare la sola fase di preprocessing sul file `main.c`, secondo le specifiche dell' ANSI C. Il file ottenuto dal preprocessing deve chiamarsi `main.E`.

Dato il seguente programma in linguaggio C, scritto nel file `prova.c` di seguito riportato, supponendo di compilare il programma usando i seguenti comandi:

```
gcc -C -ansi -Wall -pedantic -DDEBUG prova.c
```

```
gcc -o prova prova.o
```

che valore assume la variabile `k` appena prima della terminazione del programma?

```
#include <stdlib.h>
#include <unistd.h>
void main(void) {
    int k=0;
#ifdef DEBUG
    k=13;
#else
    k=17;
#endif
    /* valutare qui il valore di k */
    return;
}
```

-----

Siano dati i seguenti files, il cui nome e' indicato nei commenti:

```
/* var.h */
```

```
int A=10;
```

```
/* func.c */
```

```
#include "var.h"
int func(void) {
    return(A+3);
}
```

```
/* main.c */
```

```
#include "var.h"
extern int fun(void);
int main(void) {
    A=fun();
}
```

Supponiamo di eseguire I seguenti comandi per la compilazione e linking del programma:

```
gcc-ansi -c -Wall -pedantic func.c
```

```
gcc-ansi -c -Wall -pedantic main.c
```

```
gcc-ansi -o main.exe -Wall -pedantic func.o main.o
```

- La compilazione di `func.c` va a buon fine o genera errori? Se genera errori, quali?
- La compilazione di `main.c` va a buon fine o genera errori? Se genera errori, quali?
- Solo nel caso rispondiate che le due precedenti compilazioni vanno a buon fine, il linking dei due moduli per generare l'eseguibile `main.exe`, va a buon fine o genera un errori? Se genera errori, quali?

- Solo nel caso rispondiate che le due precedenti compilazioni e il linking vanno a buon fine, eseguendo l'eseguibile quale valore assume la variabile A al termine dell'esecuzione?

-----

Se un programma è costituito da un unico file scritto in linguaggio C, serve la fase di linking per creare il corrispondente eseguibile?

-----

161) Dato il seguente programma C, dopo compilazione e linking, quando tale programma viene eseguito, in che parte logica della memoria RAM del mio programma si trova la variabile varA ?

```
#include <stdio.h>
int varA;
void main(void) {
    int varB=10;
    varB+=varA;
    printf( "%d", varB);
}
```

162) Dato il seguente programma C, dopo compilazione e linking, quando tale programma viene eseguito, in che parte logica della memoria si trova la variabile varB ?

```
#include <stdio.h>
int varA;
void main(void) {
    int varB=10;
    printf( "%d", varB);
}
```

-----

171) Dato il programma C costituito dai due seguenti files, A.c e B.c, e ricordando che in compilazione un warning non è un errore, allora:

- a) La compilazione di A.c produce qualche errore? Se si, quale?
- b) La compilazione di B.c produce qualche errore? Se si, quale?
- c) Nel caso si risponda che la compilazione di A.c e B.c avviene senza errori, allora, il linking di A.o e B.o produce qualche errore ?
- d) Nel caso si risponda che il linking avviene senza errori, allora, durante l'esecuzione dell'eseguibile ottenuto dal linking, accade qualche errore?
- e) Nel caso si risponda che in esecuzione non avviene alcun errore, cosa stampa a video il programma ?

```
/* A.c */
#include <stdlib.h>
#include <stdio.h>
static int A=10;
int funzione(int i);
void main(void) {
    A=funzione(3);
    printf( "%d", A);
}
```

```
/* B.c */
#include <stdlib.h>
#include <stdio.h>

static int A=100;
int funzione(int i) {
    return ( A * i );
}
```

-----  
Cosa sono le system calls e a che livello dell'architettura degli elaboratori si trovano?

Cosa sono le librerie di sistema e a che livello dell'architettura degli elaboratori si trovano?

Qual è la differenza tra le librerie statiche e le librerie dinamiche (quest'ultima detta anche impropriamente shared libraries o anche dynamically loaded libraries).

Consideriamo un programma che è stato creato usando delle librerie. Al momento dell'esecuzione del programma, qual' è la differenza se il programma usa librerie statiche e le librerie dinamiche.

Consideriamo un programma che usa delle librerie. Al momento del linking del programma, qual' è la differenza se il programma usa librerie statiche e le librerie dinamiche.

Quali sono i vantaggi e gli svantaggi dell'uso delle librerie dinamiche rispetto alle librerie statiche?

A cosa serve il **loader** ?

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Descrivere **cosa contiene** e **a cosa serve** la parte del record di attivazione chiamata **dynamic link**.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Descrivere **cosa contiene** e **a cosa serve** la parte del record di attivazione chiamata **return address**.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Consideriamo una generica funzione fA che chiama una generica funzione fB. Descrivere quale parte del record di attivazione di fB viene messo dalla funzione chiamante fA, e quale parte invece viene messo dalla funzione chiamata fB stessa.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Supponiamo di avere a disposizione un processore della famiglia IA32 (x86) oppure un 8088, scegliete voi. Supponiamo di vedere il codice assembly di una funzione fB che viene chiamata da un'altra funzione fA. Qual'è la sequenza di istruzioni assembly che vengono comunque sempre eseguite per terminare la funzione chiamata fB e ritornare all'esecuzione della funzione chiamante fA?. Indicate voi eventuali altre ipotesi che ritenete necessario specificare.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Consideriamo una generica funzione fA che chiama una funzione fB, il cui prototipo è  
int32\_t fB ( int32\_t n, int32\_t g );  
Descrivere qual è il formato del record di attivazione di fB.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Supponiamo di avere a disposizione un processore della famiglia IA32 (x86) oppure un 8088, scegliete voi.

Consideriamo una funzione il cui prototipo è int16\_t fB ( int16\_t n, int16\_t g );  
Supponiamo che all'interno di quella funzione fB sia dichiarata una variabile locale int16\_t K;  
Consideriamo la traduzione di quella funzione in linguaggio assembly.  
Scrivere le istruzioni assembly, eseguite all'interno di quella funzione fB, dopo che il record di attivazione della funzione fB è stato completamente formato, che inizializzano la variabile **K** assegnandole il valore 9.

Consideriamo il linguaggio ANSI C e la convenzione di chiamata delle funzioni di tipo x86 \_\_cdecl. Supponiamo di avere a disposizione un processore della famiglia IA32 (x86) oppure un 8088, scegliete voi.

Consideriamo una funzione il cui prototipo è `int16_t fB (int16_t n, int16_t g );`

Supponiamo che all'interno di quella funzione fB sia dichiarata una variabile locale `int16_t K;`

Consideriamo la traduzione di quella funzione in linguaggio assembly.

Scrivere le istruzioni assembly, eseguite all'interno di quella funzione fB, dopo che il record di attivazione della funzione fB è stato completamente formato, che assegnano alla variabile **n** il valore **5**.

-----  
Nel processore 8088, a cosa serve il registro SP?

-----  
DOMANDE RELATIVE ALLA LEZIONE di mercoledì 3 dicembre 2014.

Nell'80888, a cosa serve il registro di segmento DS?

Nell'80888, a cosa serve il registro di segmento CS?

Suppongo di conoscere il contenuto dei registri di un processore 8088. A che indirizzo si trova la prossima istruzione macchina da eseguire?

111)

Quanti bytes può contenere al massimo un segmento di memoria nell'8088? Perché ?

112)

Perché nell'8088 ciascun segmento di memoria inizia ad un indirizzo multiplo di 16?

121)

In un 8088, se i registri di segmento contengono i seguenti valori: DS=0 SS=1 CS=2 ES=3

ed i registri IP, SP, BP, BX, DI contengono tutti il valore 2,

allora qual è l'indirizzo fisico di inizio del blocco di due bytes che viene sovrascritto da ciascuna delle seguenti istruzioni, qualora fosse eseguita isolatamente?

POP BX

POP [BX]

POP [BP]

POP [BX+2]

122)

In un 8088, se i registri di segmento contengono i seguenti valori: DS=0 SS=1 CS=2 ES=3

ed i registri IP, SP, BP, BX, DI contengono tutti il valore 2,

allora, supponendo che ciascuna delle seguenti istruzioni sia eseguita isolatamente, per ciascuna istruzione dire se:

- ciascuna delle seguenti istruzioni sovrascrive qualche locazione di memoria RAM?,

- se l'istruzione sovrascrive la memoria, qual è l'indirizzo fisico di inizio del blocco di due bytes che viene sovrascritto?

MOV BX, DI

```
MOV [BX], DI
MOV [BP], DI
MOV [BX+2], DI
```

130)

Se un programma scritto in linguaggio C implementa una funzione, e quella funzione contiene una variabile locale (ad es: int j=0;), e quella variabile ad un certo istante viene incrementata dal programma (ad es: j+=5;), e quel programma viene tradotto in linguaggio assembly dal compilatore gcc usando il flag -S, allora

nel sorgente in assembly di quel programma, quale registro di segmento viene usato per accedere alla variabile locale j nel momento in cui la si vuole incrementare ?

-----

Durante l'esecuzione di un programma, implementato in linguaggio ANSI C, ed eseguito su un processore 8088, viene chiamata una funzione func. Osserviamo la situazione dello stack nel momento in cui il record di attivazione della funzione è stato appena completamente collocato sullo stack. Supponiamo che il registro SS contenga 10000, che il registro SP contenga 996, che il registro BP contenga 1000. Supponiamo che lo stack, nella zona comprendente il record di attivazione contenga quello che è indicato nella colonna a destra della tabella qui sotto rappresentata. La colonna a sinistra, invece, indica gli indirizzi delle celle di memoria.

#### CONTENUTO IN MEMORIA

Address	Value
161005	0x01
161004	0x02
161003	0x03
161002	0x04
161001	0x05
161000	0x06
160 999	0x07
160998	0x08
160997	0x09
160996	0x0A
160995	0x0B

Cosa troviamo nel registro AX dopo l'esecuzione della seguente istruzione?

```
MOV AX , [BP-2]
```

Cosa contiene il dynamic linking del record di attivazione?

Se la funzione possiede una sola variabile locale e questa è di tipo uint16\_t, che valore ha quella variabile ?

-----

**4) Qual è la caratteristica hardware dell'8088 che comporta che tale processore possa indirizzare (utilizzare) 1 Megabyte di memoria?**

**5) In che modo l'8088 calcola un indirizzo in memoria (indirizzo compreso tra 0 e  $(2^{20}-1)$ ) partendo dal valore di un segmento (16 bit) e da un offset (16 bit)?**

- 6) Nell'8088, se il registro SS (16bit) vale 0000000000000010b ed il registro SP (16bit) vale 000000000000001b, qual è l'indirizzo fisico in memoria specificato dalla coppia di registri SS:SP?
- 17) Gli indirizzi specificati dall'istruzione `PUSH [BX+DI+2]` necessitano di essere rilocati staticamente durante il linking del programma che usa quell'istruzione? Perché?
- 20) Se ad un certo istante  $t_0$  guardiamo i valori dei registri dell'8088, e poi eseguiamo le due seguenti istruzioni, e poi guardiamo i registri, quali registri risultano modificati rispetto all'istante  $t_0$  ?
- ```
PUSH AX
POP  AX
```
- 23) A quale scopo viene usato il registro BP nell'implementazione assembly di una funzione ?

-----

DOMANDE RELATIVE ALLE LEZIONI di mercoledì 3 dicembre 2014 e venerdì 5 dicembre 2014.

- 157) Che cos'è, in generale, la rilocazione degli indirizzi ?
- 158) Che cos'è la rilocazione statica degli indirizzi ?
- 161) A cosa serve la segmentazione?
- 162) In un sistema che utilizza la segmentazione, cosa serve per individuare un indirizzo fisico in memoria?

Supponiamo che un programma venga eseguito più volte in uno stesso computer. All'inizio dell'esecuzione, una volta completato il caricamento in memoria, guardiamo il valore di ciascun registro di segmento. Per uno stesso registro di segmento, tale valore può essere ogni volta diverso o deve essere sempre lo stesso? Motivare la risposta.

AGGIUNGERNE ALTRE ANCORA

-----

### PUNTATORI e ARITMETICA DEI PUNTATORI

Il main di un programma definisce una variabile locale di tipo intero `int numero;` le assegna un certo valore maggiore di zero, poi chiama una funzione `func` che deve svolgere due compiti:

- a) Modificare la variabile `numero` aggiungendovi il valore 3;
- b) verificare se la variabile `numero` così modificata è pari o dispari, restituendo come risultato 0 se `numero` è pari, altrimenti 1.

Implementare la funzione `func`.

AGGIUNGERNE ALTRE ANCORA

-----

- 213) Se una variabile `ch` viene definita in questo modo `char ch;` e se l'indirizzo di tale variabile `ch` è 0x1000,

e ad un certo istante i valori dei bytes in memoria sono quelli rappresentati nella figura a sinistra, supponendo che dopo questo istante venga eseguita l'istruzione seguente

**\* ( (short int\*) ( ( &ch ) +2 ) ) = 0 ;**

quali saranno i valori in memoria dopo l'esecuzione dell'istruzione?  
Si scrivano i valori nella figura a destra

**PRIMA**

| Address | Value |
|---------|-------|
| 0x1005  | 0xFF  |
| 0x1004  | 0xFF  |
| 0x1003  | 0xFF  |
| 0x1002  | 0xFF  |
| 0x1001  | 0xFF  |
| 0x1000  | 0xFF  |
| 0x0FFF  | 0xFF  |
| 0x0FFE  | 0xFF  |
| 0x0FFD  | 0xFF  |
| 0x0FFC  | 0xFF  |
| 0x0FFB  | 0xFF  |

**DOPO**

| Address | Value |
|---------|-------|
| 0x1005  |       |
| 0x1004  |       |
| 0x1003  |       |
| 0x1002  |       |
| 0x1001  |       |
| 0x1000  |       |
| 0x0FFF  |       |
| 0x0FFE  |       |
| 0x0FFD  |       |
| 0x0FFC  |       |
| 0x0FFB  |       |

-----

Il compilatore ANSI C riesce a compilare e linkare questo programma oppure genera qualche errore?

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
int main(void) { int k, m=20; for( k=(int)'a'; k<(int)'z'; k++) m+=3; return(0); }
```