

**Corso di Ingegneria degli Algoritmi**  
**Facoltà di Ingegneria, Sapienza Università di Roma**  
**Appello del 9 febbraio 2010 (5/6 cfu) – Sessione A**

Durata: 2 ore

**Domanda 1.** Si affrontino i seguenti punti relativi al linguaggio C:

- a) Si traduca in C la seguente dichiarazione usando le regole sulla specifica di espressioni di tipo:

“ $v$  è una funzione senza argomenti che restituisce un puntatore a un array di 10 puntatori a puntatori a funzione senza parametri che restituisce un puntatore a `int`”

- b) Cosa stampa il seguente programma?

```
#define PRODOTTO(a,b) a*b
int main(){
    printf("%d\n", 2*PRODOTTO(2+3,4));
    return 0;
}
```

Motivare la risposta, discutendo le eventuali problematiche connesse con l'uso delle macro che emergono in questo esempio.

- c) Sia  $x$  una variabile di tipo `int`,  $p$  una variabile di tipo `int*` e  $q$  una variabile di tipo `int**`. Per ciascuna delle seguenti espressioni C:

1. `&(*q+x)`   2. `*(&q+*p)`   3. `&x+x`   4. `&p+p`   5. `(**q)*x`

dire se è una espressione valida e in caso affermativo determinarne il tipo (es. `int`, `int*`, ecc.) e la categoria (*Lvalue* o *Rvalue*).

**Domanda 2.** Qual è il minimo numero di accessi a disco necessari nel caso peggiore a qualunque algoritmo di ordinamento per ordinare un file di  $N$  item usando memoria interna di dimensione  $M$  e dimensione di blocco  $B$ ? Motivare la risposta fornendo una dimostrazione completa e dettagliata del lower bound.

**Domanda 3.** Si discuta il funzionamento dei seguenti due tipi di liste, discutendone scopi, caratteristiche e fattore di carico:

- a) Liste unrolled
- b) Liste XOR

Ai fini del calcolo del fattore di carico, denotare con  $n$  il numero di item memorizzati nella lista,  $k$  il numero di byte per item,  $h$  il numero di byte per blocco di memoria heap usati internamente dall'allocatore (per la header, ecc.) e  $p$  il numero di byte necessari per rappresentare un puntatore. Fare un esempio istanziando i valori di  $p$ ,  $k$ , e  $h$  (es.  $p=4$ ,  $k=2$  ecc.).