

Fondamenti di Informatica A-L – (Prof. P. Zingaretti)

PROVA SCRITTA – 19 aprile 2004

Avvertenze:

- * Consegnare **solo fogli formato A4**
- * **Scrivere su un solo lato** (no fronte-retro)
- * In ordine di preferenza usare: 1) **inchiostro nero**; 2) matita; 3) inchiostro rosso; 4) inchiostro blu
- * **In testa a ciascun foglio** scrivere: cognome, nome, numero progressivo di pagina rispetto al totale
esempio per il secondo foglio di 3 consegnati: Giuseppe Russo 2/3
- * Mantenere sul banco il **libretto o altro documento di riconoscimento** fino a controllo avvenuto
- * Nient'altro deve trovarsi sul banco: **non è consentito consultare libri, dispense, appunti, ecc.**
- * La **correzione** di riferimento per l'autovalutazione verrà effettuata in questa stessa aula alle ore **15:00**
- * La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- * La **prova orale** si terrà **giovedì 22 aprile** alle ore **9:00** presso il D.I.I.G.A.



Si ricorda che chi si presenterà all'orale **DEVE** portare l'implementazione al computer della propria soluzione, eventualmente corretta, **corredata di tutto quanto necessario alla verifica** del corretto funzionamento.

1. (18 punti)

Si definisca una funzione C che costruisce un albero binario **bilanciato** con gli elementi di un **vettore di interi**.

2. (4 punti)

Si dichiarino le seguenti variabili nel linguaggio C:

A è un puntatore ad un array di 12 interi

B è un array di 12 array di 3 puntatori a intero

C è un puntatore a puntatore ad una funzione che ritorna un puntatore ad int

3. (4 punti)

Cosa stampa il seguente frammento di codice?

```
int x=1,y=3,*p=&x; do if (*p) { x+=*p; p++; y-- } while (y);  
printf("x=%d",x);
```

4. (4 punti)

Se f1 e f2 sono definite nel seguente modo:

```
int f1(int x, int y) { return (x-y); }
```

```
int f2(int *x, int y) { *x+=y; return (*x*y); }
```

Cosa stampa il seguente frammento di codice?

```
int a=5, b=3; printf("\nx=%d", f1(f2(&a, f1(a, b))), a);
```

1. (18 punti)

Si definisca una funzione C che costruisce un albero binario **bilanciato** con gli elementi (numero variabile) di un **vettore di interi**.

```
typedef struct btree_tag { // 1 punto
    int value; struct btree_tag *lx; struct btree_tag *rx;
} btree;

int get_nodi(btree *root) { // 6 punti
    if (root_ptr == NULL) return 0;
    else return 1 + get_nodi(root->lx) + get_nodi(root->rx);
}

void insert(btree ** root_pp, int value) { // 6 punti
    int left, right;
    while(*root_pp!=NULL) {
        left = get_nodi((*root_pp)->lx);
        right = get_nodi((*root_pp)->rx);
        if (left > right) root_pp = &((*root_pp)->rx);
        else root_pp = &((*root_pp)->lx);
    }
    *root_pp = (btree *)malloc(sizeof(btree));
    (*root_pp)->lx = NULL;
    (*root_pp)->rx = NULL;
    (*root_pp)->value = value;
}

void ins_bilanciato(int V[], int dim, btree **btp) { // 5 punti
    int i;
    for (i=0; i<dim; i++) btree_ord_ins(btp, V[i]);
}
```

2. (4 punti)

```
int (*A)[12]; // 1 punto
int *B[12][3]; // 1 punto
int (**C)(); // 2 punti
```

3. (4 punti)

Cosa stampa il seguente frammento di codice?

```
int x=1,y=3,*p=&x; do if (*p) { x+=*p; p++; y--} while (y);
printf("x=%d",x);
```

L'effetto dell'istruzione p++ è imprevedibile (errore) e quindi non si sa cosa verrà stampato (ammesso che il programma giunga alla printf).

4. (4 punti)

x=9