

Fondamenti di Informatica I

PROVA SCRITTA – 21 gennaio 2002

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
- * La **correzione** di riferimento per l'autovalutazione verrà effettuata in questa stessa aula alle ore **16:45**
- * La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- * La **prova orale** si terrà alle ore **9:30** presso l'Istituto di Informatica



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. (4 punti)

Si scriva la funzione C che calcola la somma dei valori contenuti su un albero binario di numeri interi.

2. (8 punti)

Utilizzando la funzione dell'esercizio precedente, si scriva la funzione C che appende (cioè inserisce un nodo come foglia) un valore intero su un albero binario di numeri interi in modo da minimizzare la differenza di "peso" tra i sottoalberi destro e sinistro di tutti (ossia dal padre alla root) i suoi progenitori.

3. (10 punti)

Si definisca una funzione C che **crea dinamicamente** la matrice M di dimensioni RxC, con R e C forniti come parametri, e **la restituisce con ciascun elemento inizializzato** al valore dato dal prodotto tra il corrispondente indice di riga e quello di colonna; ad esempio, l'elemento M[5][12] verrà inizializzato a 60 (=5 * 12).

4. (8 punti)

Si dica cosa esprimono le seguenti dichiarazioni in linguaggio C:

```
int *x[5][10];  
struct albero **y;  
int *(*f)(char *a[]);  
void g(int (*b)[], int **c);
```

Esercizio n° 1 (4 punti)

```
struct btree{
    int value;
    struct btree * lx;
    struct btree * rx;
};

int get_sum(struct btree * root) {
    if(root==NULL) return 0;
    else return root->value+get_sum(root->lx)+get_sum(root->rx);
}
```

Esercizio n° 2 (8 punti)

```
void insert(struct btree ** root, int value) {
    int left_sum, right_sum;

    while(*root!=NULL) {
        left_sum=get_sum((*root)->lx);
        right_sum=get_sum((*root)->rx);
        if(left_sum>=right_sum) root=&((*root)->lx);
        else root=&((*root)->rx);
    }
    *root=(struct btree *)malloc(sizeof(struct btree));
    (*root)->lx=NULL;
    (*root)->rx=NULL;
    (*root)->value=value;
}
```

Esercizio n° 3 (10 punti)

```
int *matrix(int R, int C) {
    int i, j, *M;

    M=(int *)malloc(R*C*sizeof(int));
    for (i=0; i<R; i++)
        for (j=0; j<C; j++)
            *(M+i*C+j)=i*j;
    return M;
}
```

Esercizio n° 4 (8 punti)

x è un'array di 5 array di 10 puntatori a interi;

y è un puntatore a un puntatore alla struttura albero;

f è un puntatore a una funzione che ha un'array di puntatori a char come parametro e ritorna un puntatore a un intero;

g è una funzione void che ha 2 puntatori a puntatori a int come parametri