

Fondamenti di Informatica I

PROVA SCRITTA – 22 marzo 2001

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 **13:45**
- * La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- * La **prova orale** si terrà **mercoledì 28 marzo** alle ore **14:30** in aula **160/4**
- * Si ricorda che chi si presenterà all'orale deve portare la completa implementazione al computer della propria soluzione, eventualmente corretta.

1. (6 punti)

Si definisca la struttura dati (colonna) adeguata alla rappresentazione di sequenze di 13 caratteri con associato, a ciascuna sequenza, un valore intero (punteggio). Si scriva la funzione c (Pascal) che:

- 1) riceve in input un insieme di dati colonna e una sequenza C di 13 caratteri, scelti tra '1', 'x' e '2';
- 2) per ogni sequenza dell'insieme, calcola e memorizza nel punteggio il numero dei caratteri uguali con quelli della sequenza C, a parità di posizione;
- 3) ritorna il punteggio massimo trovato per una sequenza.

2. (18 punti)

Dati 2 alberi binari, si scriva la funzione c (Pascal) che:

- 1) ritorna l'esito (true / false) della ricerca sul primo albero binario di un nodo che abbia un certo valore, fornito in input alla funzione;
- 2) se la ricerca ha esito positivo cancella il (libera la memoria del) sotto-albero relativo al nodo trovato;
- 3) sostituisce il secondo albero fornito in input al posto del nodo cancellato.

Suggerimento: fare attenzione a come ritornare l'eventuale nodo trovato in 1) per risolvere correttamente 2) e 3).

3. (6 punti)

Si esprimano le seguenti dichiarazioni di variabile in linguaggio c (Pascal):

X e' un array di 10 array di 20 variabili double

Y e' un puntatore ad array di 10 variabili intere

Z e' un puntatore a 10 variabili double

W e' un array di 10 puntatori ad array di 20 variabili intere

ESERCIZIO 1:

```

struct colonna {int punteggio; char C[13]};

int calcola_punteggio_max(struct colonna *concorso, unsigned long n, char *C);

int calcola_punteggio_max(struct colonna *concorso, unsigned long n, char *C) {
    int i,j,max=0;
    for (i=0;i<n;i++) {
        concorso[i].punteggio=0;
        for (j=0;j<13;j++)
            if (concorso[i].C[j]==C[j])
                concorso[i].punteggio++;
        if (max < concorso[i].punteggio) max=concorso[i].punteggio;
    }
    return max;
}

```

ESERCIZIO 2:

```

struct btree {int val; struct btree *dx_ptr; struct btree *sx_ptr};

int bt_merge (struct btree **bt1_ptr_ptr, struct btree *bt2_ptr, int val);
void free_sbt (struct btree *bt_ptr);

int bt_merge (struct btree **bt1_ptr_ptr, struct btree *bt2_ptr, int val) {
    if ((*bt1_ptr_ptr)->val == val) {
        struct btree *bt_ptr_app;

        bt_ptr_app=*bt1_ptr_ptr; //scambia
        *bt1_ptr_ptr=bt2_ptr;
        free_sbt(bt_ptr_app);
        return TRUE;
    }
    else {
        struct btree **bt_ptr_ptr_app;

        bt_ptr_ptr_app=bt1_ptr_ptr; //salvo il padre per ricorsione dx

        if (((*bt1_ptr_ptr)->sx_ptr != NULL) &&
            bt_merge(&(*bt1_ptr_ptr)->sx_ptr, bt2_ptr, val)) return TRUE;

        *bt1_ptr_ptr=*bt_ptr_ptr_app;
        if (((*bt1_ptr_ptr)->dx_ptr != NULL) &&
            bt_merge(&(*bt1_ptr_ptr)->dx_ptr, bt2_ptr, val)) return TRUE;
    }
    return FALSE; //nodo val non trovato
}

void free_sbt(struct btree *bt_ptr){
    if (bt_ptr->sx_ptr != NULL) free_sbt(bt_ptr->sx_ptr);
    if (bt_ptr->dx_ptr != NULL) free_sbt(bt_ptr->dx_ptr);
    free(bt_ptr);
}

```

ESERCIZIO 3:

```

double X[10][20];
int (*Y)[10];
double *Z;
int (*W[10])[20];

```