

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

PROVA SCRITTA – 15 gennaio 2004

Avvertenze:

- * Consegnare **solo fogli formato A4**
- * **Scrivere su un solo lato** (no fronte-retro)
- * **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
- * 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 **15:30**
- * La consegna delle **fotocopie** dei compiti avverrà **in questa stessa aula** al termine della correzione
- * La **prova orale** si terrà **giovedì 22 gennaio** 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. (10 punti)

Si scriva una funzione C che riceve in input un numero naturale (intero senza segno) rappresentato come stringa di cifre decimali e restituisce in output la corrispondente stringa binaria

Esempio: "1025" diventa "10000000001"

2. (11 punti)

Si scriva una funzione C che riceve in input un generico albero binario in cui a ciascun nodo è associato un valore intero memorizzato nel campo val.

La funzione restituisca in output un nuovo (cioè quello originale non deve essere modificato) albero binario identico all'originale nella struttura, ma tale che per ogni nodo i valori memorizzati nei campi val contengano la **somma** di tutti i campi val dell'albero (sotto-albero dell'originale) di cui quel nodo stesso è radice.

Esempio: 7(4(3, 11), 9(12(1, 25), 21)) diventa 86(18(3, 11), 68(38(1, 25), 21))

3. (5 punti)

Se f1 e f2 sono definite nel seguente modo:

```
int f1(int *x) { return *x>(*x)++; }  
void f2(int *x, int **y) { *x=f1(*y); }
```

Cosa stampa il seguente frammento di codice?

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

Nota: oltre al risultato riportare (magari graficamente) anche il ragionamento effettuato.

4. (4 punti)

Se f1 è definita come: void f1(int a, int *b) { int i=0; for (i=0; i<a; i+=2) *b+=a++; }

Cosa stampa il seguente frammento di codice?

```
int a=2, *b=&a; f1(a+1, b); printf("\nx=%d, y=%d", a, *b);
```

Nota: oltre al risultato riportare (magari graficamente) anche il ragionamento effettuato.

1. (10 punti)

```
char * str_dec2bin(char *dec) {
    unsigned long val = 0;
    int i = 0, j, b=1;    char *bin=(char *)malloc(100);
    while (dec[i]) i++;    //i=strlen(dec)=lunghezza stringa decimale
    for (j=i-1; j>=0; j--) {
        if ((dec[j] >= '0')&&(dec[j] <= '9'))
            val+=b*(dec[j]-'0');    //
        else return NULL;    //esco per errore
        b*=10;    //aggiorno peso
    }
    bin[0]='\0';
    j=0;
    while (val > 0) {    for(i=j; i>=0; i--)
        bin[i+1] = bin[i];    //faccio scorrere tutti gli elementi
    bin[0] = (val%2)+'0';    //aggiungo in testa carattere 0 o 1    val
    /= 2;
        j++;    //incremento lunghezza stringa    }    return bin;}
}
```

2. (11 punti)

```
typedef struct bt {int val; struct bt *sx; struct bt *dx;} btree;

btree * modify_bt(btree *bt) {
    btree *tmp=NULL;
    if (bt!=NULL) {
        tmp=(btree *)malloc(sizeof(btree)); //creo copia
        tmp->sx=modify_bt(bt->sx); //collego figlio sx
        tmp->dx=modify_bt(bt->dx); //collego figlio dx
        tmp->val=bt->val; //val radice
        if (tmp->sx) tmp->val+=tmp->sx->val; //+ val sottoalbero sx
        if (tmp->dx) tmp->val+=tmp->dx->val; //+ val sottoalbero dx
    }
    return tmp;
}
```

3. (5 punti)

x=25, y=36

4. (4 punti)

x=14, y=14