

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

PROVA SCRITTA – 12 dicembre 2005

### Avvertenze:

- \* Consegnare **solo fogli formato A4 scritti 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
- \* 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 **17:45**
- \* La consegna delle **fotocopie** dei compiti avverrà al termine della correzione
- \* La **prova orale** si terrà **lunedì 19 dicembre** alle **ore 13:30** 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. (15 punti)

Si definisca una funzione C che restituisce la stringa con caratteri ordinati ottenuta dalla fusione di due stringhe (non ordinate) di lunghezza variabile, ricevute in input.

Esempio: "studente" e "ammesso" devono fornire in uscita la stringa "adeeemmnosssttu".

### 2. (8 punti)

Dato un albero binario di ricerca in cui il valore intero memorizzato in ciascun figlio sinistro (destro) è minore (maggiore) di quello memorizzato nel rispettivo nodo padre, si scriva una funzione C che riceve in input tale albero e lo trasforma in uno con ordinamento invertito, cioè con il valore intero memorizzato in ciascun figlio sinistro (destro) maggiore (minore) di quello memorizzato nel rispettivo nodo padre.

**N.B.:** non si deve creare un nuovo albero, cioè non si può usare l'allocazione dinamica (altrimenti il punteggio massimo è di 2 punti)

### 3. (7 punti)

Date le funzioni f1 ed f2 definite come:

```
int f1(int **a) { int *b=*a+1; while (**a) *a[0]+=f2(b); return ++*b; }
int f2(int *a) { *(a+1)*=-1; return --*a**(a+1); }
```

cosa stampa il seguente frammento di codice?

```
int a[3]= {2, 5, 1}, *b=a; printf("\n%d, ", f1(&b));
```

**N.B.:** se non si riporta graficamente il ragionamento effettuato il punteggio massimo è di 3 punti.

**1. (15 punti)**

```
char *merge_str(char *str_in1, char *str_in2) {
    char *str_out;
    int i, len=0, len1=0, len2=0;

    while (str_in1[len1]!='\0') len1++; //len=strlen(str_in1);
    while (str_in2[len2]!='\0') len2++; //len=strlen(str_in2);
    str_out=(char *)malloc(len1+len2+1); //len+EOLN
    for (i=0; i<len1; i++) str_out[len++]=str_in1[i];
    for (i=0; i<len2; i++) str_out[len++]=str_in2[i];
    str_out[len]='\0'; //termino nuova stringa

    bubbleSort(str_out, len);

    return str_out;
}
```

**2. (8 punti)**

```
typedef struct btree_tag {
    int value; struct btree_tag *left; struct btree_tag *right;
} btree;

void inverti_ordine (btree *btp) {
    btree *tmp;
    if (btp!=NULL) {
        tmp=btp->left; btp->left=btp->right; btp->right=tmp;
        inverti_ordine(btp->left);
        inverti_ordine(btp->right);
    }
}
```

**3. (7 punti)**

