

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

PROVA SCRITTA – 10 dicembre 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
- \* 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 **16:00**
- \* La consegna delle **fotocopie** dei compiti avverrà al termine della correzione
- \* La **prova orale** si terrà **martedì 17 dicembre** alle ore **9:00** 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. (8 punti)

Si definiscano due funzioni C, una in forma iterativa e l'altra in forma ricorsiva, che calcolano la lunghezza di una lista di interi in rappresentazione collegata mediante puntatore.

### 2. (10 punti)

Definire una funzione C che costruisce un albero binario di ricerca con gli elementi di un vettore di interi, provvedendo ad eliminare i valori doppi e restituendo il numero di nodi creati.

### 3. (12 punti)

Sia dato il file di testo "libri.txt" in cui ciascuna delle righe dovrebbe contenere (se non ci sono errori) nell'ordine: anno di pubblicazione, autori, titolo, casa editrice e prezzo di un libro. Il carattere ';' funge da separatore tra queste 5 informazioni ma può essere preceduto e seguito da spazi. L'anno di pubblicazione è nel formato a 4 cifre, il prezzo (espresso in dollari) è preceduto dal carattere '\$' e approssimato a 2 cifre decimali, mentre gli altri campi sono di lunghezza variabile e possono contenere spazi. Quale esempio, di seguito si mostrano 6 righe di questo file, di cui solo le prime 3 sono corrette, mentre le altre 3 vanno saltate:

```
2001; A. Autore1 ; titolo testo 1; casa editrice 1; $35.00
2002 ; B. Autor, C. Autor; titolo del secondo libro; editore ; $55.50
2001; A. Autore1 ; $35.00
2002 ; B. Autor, C. Autor; titolo del secondo libro; editore ;
2001; A. Autore1 ; titolo testo 1; casa editrice 1; 35.00
B. Autor, C. Autor; titolo del secondo libro; editore ; $55.50
```

Si definisca una funzione C che considera solo le righe che contengono quantomeno i campi "anno" e "prezzo" e che restituisce sia il numero di libri pubblicati in un determinato anno (fornito in input) che il loro prezzo medio.

### Suggerimenti:

- 1) la funzione standard: `char *strrchr(const char *s1, int c);` ritorna un puntatore all'ultima occorrenza di c nella stringa s1 o NULL se il carattere c non appartiene alla stringa.
- 2) ricordarsi che la funzione `fscanf()` effettua la lettura di un campo nelle righe successive (i.e., oltre il \n) se non lo trova nella riga corrente.

## 1. (8 punti)

Si definiscano due funzioni C, una in forma iterativa e l'altra in forma ricorsiva, che calcolano la lunghezza di una lista di interi.

```
typedef struct list_tag {int value; struct list_tag *next;} list;

int lengthI(list *lp) { //versione iterativa
    int length=0;
    while (lp!=NULL) {
        length++;
        lp=lp->next;
    }
    return length;
}

int lengthR(list *lp) { //versione ricorsiva
    if(lp==NULL) return 0;
    else return 1+lengthR(lp->next);
}
```

## 2. (10 punti)

Definire una funzione C che costruisce un albero binario di ricerca con gli elementi di un vettore di interi, provvedendo ad eliminare i valori doppi e restituendo il numero di nodi creati.

```
typedef enum {FALSE, TRUE} Boolean;
typedef struct btree_tag {
    int value;
    struct btree_tag *left;
    struct btree_tag *right;
} btree;

void ins_ord(int V[], int dim, btree **btp, int *nnodi) {
    int i;
    for (i=0; i<dim; i++)
        *nnodi+=btree_ord_ins(btp, V[i]);
}

Boolean btree_ord_ins(btree **btp, int value) {
    while (*btp!=NULL) //posizionamento
        if ((*btp)->value == value) return FALSE;
        else if ((*btp)->value > value)
            btp=&((*btp)->left);
        else btp=&((*btp)->right);
    //inserimento
    *btp=(btree *)malloc(sizeof(btree));
    (*btp)->value=value;
    (*btp)->left=NULL;
    (*btp)->right=NULL;
    return TRUE;
}
```

### 3. (12 punti)

Sia dato il file di testo "libri.txt" in cui ciascuna delle righe dovrebbe contenere (se non ci sono errori) nell'ordine: anno di pubblicazione, autori, titolo, casa editrice e prezzo di un libro. Il carattere ';' funge da separatore tra queste 5 informazioni ma può essere preceduto e seguito da spazi. L'anno di pubblicazione è nel formato a 4 cifre, il prezzo (espresso in dollari) è preceduto dal carattere '\$' e approssimato a 2 cifre decimali, mentre gli altri campi sono di lunghezza variabile e possono contenere spazi. Quale esempio, di seguito si mostrano 6 righe di questo file, di cui solo le prime 3 sono corrette, mentre le altre 3 vanno saltate:

```
2001; A. Autore1          ; titolo testo 1;          casa editrice 1; $35.00
 2002 ; B. Autor, C. Autor; titolo del secondo libro; editore ;    $55.50
2001; A. Autore1          ; $35.00
 2002 ; B. Autor, C. Autor; titolo del secondo libro; editore ;
2001; A. Autore1          ; titolo testo 1;          casa editrice 1; 35.00
  B. Autor, C. Autor; titolo del secondo libro; editore ;    $55.50
```

Si definisca una funzione C che considera solo le righe che contengono quantomeno i campi "anno" e "prezzo" e che restituisce sia il numero di libri pubblicati in un determinato anno (fornito in input) che il loro prezzo medio.

```
typedef struct {
    int anno;
    int num_libri;
    float p_medio;
} result;

result report(char *filename, int year) {
    FILE* fp;
    int anno;
    float prezzo;
    char buf[100], *p;
    result rep={year, 0, 0.0F}; //inizializzazione

    if ((fp=fopen(filename, "r"))==NULL) {
        perror("Il file non esiste!"); exit(1);
    }
    while (!feof(fp)) {
        fgets(buf,199,fp); //leggo una linea
        if ((sscanf(buf,"%d", &anno)==1)    && //esiste anno
            (anno==year)                   && //è l'anno cercato
            ((p=strrchr(buf,'$'))!=NULL)   && //esiste '$'
            (sscanf(p+1,"%f", &prezzo)==1) //esiste prezzo
        ) { //se tutti i controlli sono OK aggiungo in rep
            rep.num_libri++;
            rep.p_medio+=prezzo;
        } //else legge prossima linea
    } //end of while (!feof(fp))
    fclose(fp);
    if (rep.num_libri) rep.p_medio/=rep.num_libri;
    return rep;
}
```