

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

PROVA SCRITTA – 11 settembre 2003

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 **10:15**
- * La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- * **Prova orale giovedì 18 settembre** alle ore **9:00** c/o il Dip. Ing. Informatica, Gestionale e dell'Automazione.



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 definisca una funzione C che riceve in input due stringhe di lunghezza variabile e restituisce la stringa costruita alternando, finché possibile, un carattere della prima stringa con uno della seconda e quindi aggiungendo i caratteri residui della stringa più lunga, come nell'esempio seguente: str1="ABCDE", str2="123", str_out="A1B2C3DE".

2. (10 punti)

Si definisca una funzione C che costruisce un albero binario di ricerca con gli elementi di un vettore di interi.

3. (10 punti)

Si definisca in C la struttura dati "libretto" idonea a memorizzare soltanto gli esami sostenuti da uno studente (non tutti quelli sostenibili). In particolare, per ciascun esame sostenuto si memorizzino le seguenti informazioni: il nome, la data e il voto ottenuto.

Si definisca poi una funzione C che scrive su un file di testo, uno per riga, tutti i dati relativi al libretto ricevuto in input.

1. (10 punti)

```
void merge_str(char *str1, char *str2, char *str_out) {
    int i, j, k=0, len1=0, len2=0;
    while (str1[len1]!='\0') len1++;
    while (str2[len2]!='\0') len2++;
    str_out=(char *)malloc(len1+len2+1);
    if (len1>len2) j=len2; else j=len1; //j = len stringa + corta
    for (i=0; i<j; i++) { //parte da alternare
        str_out[k++]=str1[i];
        str_out[k++]=str2[i];
    }
    if (j==len1) //aggiungo restanti caratteri da str2
        for (i=len1; i<len2; i++) str_out[k++]=str2[i];
    else //aggiungo restanti caratteri da str1
        for (i=len2; i<len1; i++) str_out[k++]=str1[i];
    str_out[k]='\0';
}
```

2. (10 punti)

```
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 i;
    for (i=0; i<dim; i++) btree_ord_ins(btp, V[i]);
}

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

3. (10 punti)

```
typedef struct lib { char corso[40]; char data[10]; int voto;
                    struct lib *next;} libretto;

void print_lib(libretto *lib, char *nome_file) {
    FILE *fp=fopen(nome_file, "wt");
    int i=0;
    while (lib) {
        fprintf(fp, "%40s %10s %2d\n", lib->corso, lib->data, lib->voto);
        lib=lib->next;
    }
    fclose(fp);
}
```