

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

PROVA SCRITTA – 8 aprile 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 **15:45**
- * La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- * La **prova orale** si terrà **lunedì 14 aprile** alle ore **14:30** 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. (10 punti)

Si definisca una funzione C (Pascal) che riceve in input un valore intero con segno e restituisce la corrispondente stringa in **complemento a 2 su 32 bit**.

2. (9 punti)

Per ogni tipo di articolo presente in magazzino si considerino i seguenti attributi: codice, numero pezzi, prezzo di acquisto medio. Si definisca in linguaggio C (Pascal) una funzione che inserisce un tipo di articolo su un **albero binario di ricerca** in base al numero di pezzi.

3. (7 punti)

Si definisca una funzione C (Pascal) che riceve in input una **stringa di lunghezza variabile** e restituisce la stringa invertita (ad esempio, la stringa invertita di "esempio" è "oipmese").

4. (4 punti)

Si esprimano le seguenti **dichiarazioni** in linguaggio C (Pascal):

A è una procedura con una matrice tridimensionale di interi come parametro

B è una funzione che ritorna un puntatore a un array di interi

```

char *compl2(int N) {
    int i=31, neg=0;
    char *str=(char *)malloc(33); //32 bit + EOLN

    str[32]='\0'; //terminazione stringa
    if (N<0) {neg=1; N=-N;} //N=abs(N);
    while (N) { //alg. div. successive
        str[i--]=(N%2)?'1':'0'; N>>=1; //N/=2; ossia N=N/2;
    }
    for (; i>=0; i--) str[i]='0';
    if (neg) { //necessario complemento a 2
        // 001001 -> compl. -> 110110 -> +1 -> 110111
        // 001100 -> compl. -> 110011 -> +1 -> 110100
        //ossia compl. fino ad ultimo '1' escluso!!!!
        i=31;
        while (str[i]=='0') i--; //find last '1'
        for (--i; i>=0; i--) str[i]=(str[i]=='0')?'1':'0';
    }
    return str;
};

typedef struct { int cod; int num_pz; int p_medio; } articolo;
typedef struct bt_tag {
    articolo info; struct bt_tag *sx; struct bt_tag *dx; } btree;

void insert(btree **btp, articolo item) {
    while (*btp!=NULL) //posizionamento
        if ((*btp)->info.num_pz > item.num_pz)
            btp=&((*btp)->sx);
        else btp=&((*btp)->dx);
    //inserimento
    *btp=(btree *)malloc(sizeof(btree));
    (*btp)->info=item;
    (*btp)->sx=NULL;
    (*btp)->dx=NULL;
}

char *inverti_str(char *str_in) {
    int len=0, i=0;
    char *str_out;

    while (str_in[len]!='\0') len++; //len=strlen(str_in);
    str_out=(char *)malloc(len+1); //len+EOLN
    str_out[len]='\0'; //terminazione stringa
    for (i=0; i<len; i++) str_out[i]=str_in[len-1-i];
    return str_out;
}

void A(int [T][R][C]); oppure: void A(int *, int T, int R, int C);
int *B( );

```