

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

PROVA SCRITTA – 13 gennaio 2005

Avvertenze:

- * Consegnare **solo fogli formato A4** e 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
- * 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 **15:30**
- * La consegna delle **fotocopie** dei compiti avverrà al termine della correzione
- * La **prova orale** si terrà **giovedì 20 gennaio** alle ore **8:45** in aula ***vedere pagina web corso***.



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. (22 punti)

Si definisca una funzione C che **trasforma** la lista ordinata di interi ricevuta in input, nella rappresentazione collegata mediante puntatori, in una lista ordinata in senso contrario.

Ad esempio: [1, 5, 7, 23, 28] diventa [28, 23, 7, 5, 1] e [1, 5, 7, 28] diventa [28, 7, 5, 1]

NB1: per "trasforma" s'intende che **non** vengono effettuate malloc né di nuovi elementi della lista né di altre strutture dati, bensì occorre modificare opportunamente solo i campi int degli elementi della lista.

NB2: una implementazione corretta basata su un algoritmo di ordinamento verrà valutata **16 punti**.

2. (4 punti)

Cosa stampa il seguente frammento di codice:

```
int f1(int *x, int **y) { return *x**x***y; }
void f2(int *x, int **y) { *x=f1(*y,y); }
int a=3, b=2, *p=&b;
f2(&a, &p);
printf("\nx=%d, y=%d", a, f1(p,&p));
```

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

3. (4 punti)

Cosa stampa il seguente frammento di codice:

```
int f1(int *x, int *y) { int z, k=0; for (z=0; z < 2; z++) k+=(*x++)+(*y--); return k; }
int x[4]={3, 2, 5, 1};
printf("\n %d; %d", f1(x,&x[3]), f1(&x[1], &x[3]));
```

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

1. (20 punti)

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

```
void trasforma(list *first) {  
    list *tmp=first, *last=NULL;  
    int val;  
  
    while (first!= last) {  
        tmp=first;  
        while (tmp->next != last)  
            tmp = tmp->next;  
        if (tmp == first) //uscita  
            last=first;  
        else {                //scambia e aggiorna puntatori  
            val=first->val;  
            first->val=tmp->val;  
            tmp->val=val;  
            first=first->next; //aggiorno primo elemento  
            last=tmp;          //aggiorno ultimo elemento  
        }  
    }  
}
```

```
void trasforma_BS(list *first) {  
    list *tmp=first, *last=NULL;  
    int val, ordinato=0;  
  
    while (!ordinato) {  
        ordinato=1;  
        tmp=first;  
        while (tmp && tmp->next) {  
            if (tmp->val < tmp->next->val) {  
                val=tmp->val; //scambia  
                tmp->val=tmp->next->val;  
                tmp->next->val=val;  
                ordinato=0;  
            }  
            tmp = tmp->next;  
        }  
    }  
}
```

2. (3 punti)

x=8, y=8

3. (3 punti)

11, 13