

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

PROVA SCRITTA – 23 giugno 2004

### 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 **15:00**
- \* La consegna delle **fotocopie** dei compiti avverrà presso il D.I.I.G.A. al termine della correzione
- \* La **prova orale** si terrà **mercoledì 30 aprile** alle ore **9:00** 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. (20 punti)

Si definisca una funzione C che restituisce il vettore di interi ricevuto in input dopo averlo ordinato (si può usare qualsiasi algoritmo di ordinamento – **8 punti**) come nell'esempio sotto riportato, ossia in serie crescenti di valori ma senza duplicati, i quali, se presenti, danno inizio ad una nuova serie all'interno del vettore.

Esempio: 4 3 7 2 3 5 3 2 4 1 diventa 1 2 3 4 5 7 2 3 4 3 in cui si hanno le 3 serie ordinate (1 2 3 4 5 7) (2 3 4) e (3).

### 2. (5 punti)

Cosa stampa il seguente frammento di codice?

```
int f1(int *x) { int y, z=5;
                for (y=3; y; y--) if (z > y) z--; else (*x)++;
                return z;
            }
int x=5, y=3;
x=f1(&y);
printf("\nx=%d; y=%d", x, y);
```

### 3. (5 punti)

Cosa stampa il seguente frammento di codice?

```
int f1(int *x) { int y, z=2;
                for (y=0; y < 3; y++) z+=*(x++);
                return z;
            }
int x[5]={3, 2, 5, 1, 6};
printf("\n%d; %d", f1(x), f1(&x[2]));
```

### 1. (20 punti)

```
void split(int *A, int n, int start) {
    int i, j, last=n;
    for (i=start; i<last; i++)
        if (A[i]==A[i-1]) {
            for (j=i+1; j<n; j++) A[j-1]=A[j];
            A[n-1]=A[i-1]; //copio in fondo (A[i] è perso!)
            last--; //un elemento in meno da processare
            i--; //riprocesso la stessa posizione
        }
    last++;
    if (last < n) //ho elementi per una nuova serie
        split(A, n, last);
}
```

```
void disponi(int *A, int n) {
    bubblesort (A, n); // 8 punti
    split(A, n, 1);
}
```

### 2. (5 punti)

x=2; y=3

### 3. (5 punti)

12; 14