

# Fondamenti di Informatica

## Prof Aldo Franco Dragoni

### prova di programmazione del 20 luglio 2015

#### Avvertenze

- Consegnare solo e fogli bianchi formato A4.
- In ordine di preferenza usare inchiostro nero, matita, 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 fornita sul sito internet del Corso
- La consegna delle fotocopie dei compiti avverrà al termine della correzione
- Chi si presenta all'orale deve portare la propria soluzione, corretta ed autovalutata a penna rossa.

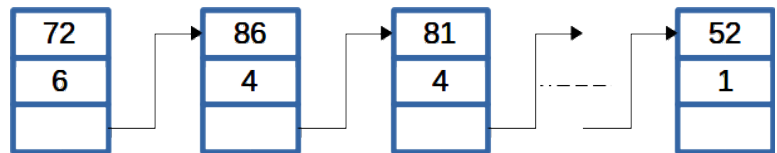
Si supponga di avere un file di testo `numeri` che contiene una successione, arbitrariamente lunga, di numeri interi, rappresentanti ciascuno il campionamento di una qualche grandezza), es:

```
63 52 81 91 82 73 28 91 82 19 72 81 94 53 19 80 72 86 72 28 61 72 82 94 38
85 73 48 73 72 83 93 84 95 28 96 77 88 99 60 71 82 73 64 75 96 87 98 29 70
92 83 94 85 26 77 58 39 50 61 72 23 54 35 96 47 68 49 70 81 92 63 84 75 86
78 89 70 41 42 23 54 35 86 77 58 69 40 51 62 13 64 55 76 87 78 96 81 97 86
```

Il programma deve:

1. caricare tutti i numeri in una lista di elementi del tipo:

```
struct elem {
    int info;
    int occorrenze;
    elem *succ;
};
```



dove `info` contiene il valore del campionamento e `occorrenze` il numero di volte in cui quel valore compare nella serie; la lista deve essere creata in maniera da mantenerla ordinata in senso decrescente rispetto al numero di `occorrenze` del valore

2. stampare a video la “*moda*” (il valore che compare più volte; a parità se ne scelga uno a caso), la *media aritmetica* dei valori, e la loro serie completa, come da esempio seguente (riferito al file `numeri` precedente):

```
prompt$ ./prova numeri
```

```
La moda del campione è 72 con 6 occorrenze
```

```
La media aritmetica è 68.33
```

```
La lista completa delle frequenze ordinata in senso decrescente è la seguente:
```

```
72 occorrenze: 6
86 occorrenze: 4
81 occorrenze: 4
96 occorrenze: 4
73 occorrenze: 4
82 occorrenze: 4
77 occorrenze: 3
70 occorrenze: 3
94 occorrenze: 3
28 occorrenze: 3
78 occorrenze: 2
```

```
87 occorrenze: 2
64 occorrenze: 2
58 occorrenze: 2
35 occorrenze: 2
54 occorrenze: 2
23 occorrenze: 2
75 occorrenze: 2
84 occorrenze: 2
63 occorrenze: 2
92 occorrenze: 2
61 occorrenze: 2
85 occorrenze: 2
83 occorrenze: 2
19 occorrenze: 2
91 occorrenze: 2
97 occorrenze: 1
76 occorrenze: 1
55 occorrenze: 1
13 occorrenze: 1
62 occorrenze: 1
51 occorrenze: 1
40 occorrenze: 1
69 occorrenze: 1
42 occorrenze: 1
41 occorrenze: 1
89 occorrenze: 1
49 occorrenze: 1
68 occorrenze: 1
47 occorrenze: 1
50 occorrenze: 1
39 occorrenze: 1
26 occorrenze: 1
29 occorrenze: 1
98 occorrenze: 1
71 occorrenze: 1
60 occorrenze: 1
99 occorrenze: 1
88 occorrenze: 1
95 occorrenze: 1
93 occorrenze: 1
48 occorrenze: 1
38 occorrenze: 1
80 occorrenze: 1
53 occorrenze: 1
52 occorrenze: 1
```

**ATTENZIONE:** come mostrato nell'esempio, il nome del file **non** deve essere chiesto all'utente ma **deve essere passato come argomento** al main al momento del lancio del programma.

La struttura deve essere la seguente:

```
fstream sondaggio;
typedef elem* lista;

bool estrai_elem_dato(.....) {
2 Punti
}

void insOrdinato(...) {
3 Punti
}
```

```
void carica(...) {
6 Punti
}
```

```
void stampa(...) {
1 Punto
}
```

```
float media(...) {
3 Punti
}
```

```
int main(int argc, char *argv[]) {
3 Punti
}
```

## SOLUZIONE

```
#include <iostream>
#include <fstream>
using namespace std;

fstream sondaggio;

struct elem {
    int info;
    int occorrenze;
    elem *succ;
};

typedef elem* lista;

bool estrai_elem_dato(lista& inz, elem& a) {
    lista p, q;
    for (q = inz; q!=0 && q->info!=a.info; q=q->succ) p = q;
    if (q == 0) return false;
    if (q == inz) inz = q->succ;
    else p->succ = q->succ;
    a = *q;
    delete q;
    return true;
}

void insOrdinato(lista& inz, elem a) {
    lista p = 0, q, r;
    for (q=inz; q!=0 && q->occorrenze > a.occorrenze; q = q->succ) p = q;
    r = new elem;
    *r = a;
    r->succ = q;
    // controlla se si deve inserire in testa
    if (q == inz) inz = r;
    else p->succ = r;
}

void carica(lista& frequenze) {
    elem a;
    while (sondaggio >> a.info) {
        if (estrai_elem_dato(frequenze, a)) a.occorrenze = a.occorrenze+1;
        else a.occorrenze = 1;
        insOrdinato(frequenze, a);
    }
}
```

```

    }
}

void stampa(lista l) {
    while (l != 0) {
        cout << l->info << " occorrenze: " << l->occorrenze << endl;
        l = l-> succ;
    }
}

float media(lista l) {
    int numerosita=0,totale = 0;
    while (l != 0) {
        totale += (l->info * l->occorrenze);
        numerosita += l->occorrenze;
        l = l-> succ;
    }
    return static_cast <float>(totale)/numerosita;
}

int main(int argc, char *argv[]) {
    sondaggio.open(argv[1],ios::in); // apre il filestream sondaggio
    lista frequenze = 0; // crea la lista vuota
    carica(frequenze); // riempie la lista prendendo i valori dal file e la mantiene ordinata
    cout << "La moda del campione è " << frequenze->info << " con " << frequenze->occorrenze << "
occorrenze" << endl;
    cout << "La media aritmetica è " << media(frequenze) << endl;
    cout << "La lista completa delle frequenze ordinata in senso decrescente è la seguente:" <<
endl;
    stampa(frequenze);
    sondaggio.close();
    return 0;
}

```