Introduzione all'Algoritmica per i Licei (C++).

2 – Massimo e minimo.

versione 13 gennaio 2015

Elio Giovannetti
Dipartimento di Informatica
Università di Torino

Il massimo di una sequenza immessa da tastiera.

Esperimento.

Vi comunico lentamente una sequenza di numeri;
voi, senza usare né carta né matita né altri strumenti,
alla fine dovete dirmi qual è il massimo degli elementi.

...

Come avete fatto per dare la risposta giusta ?

Ad ogni passo:

- avete tenuto in memoria il massimo fino a quel momento;
- avete ricevuto il nuovo numero;
- lo avete confrontato con il massimo:
  - se era maggiore lo avete memorizzato come nuovo massimo dimenticando il precedente;
  - altrimenti avete continuato a memorizzare il massimo, dimenticando il numero ricevuto.

L'algoritmo

ascolta il primo numero e memorizzalo come massimo

ripeti indefinitamente :

    ascolta ciò che vien detto (per brevità, detto)

    se detto è "ho finito": esci dal ciclo

    altrimenti:

        considera il numero espresso da detto;

        se numero > massimo: memorizza numero come massimo

fine del ciclo

scrivi o pronuncia il massimo

Nota: l'algoritmo funziona solo se termino la sequenza di numeri con la frase "ho finito"; se ad un certo punto, invece di dire un numero, dico "Buongiorno", non si sa se ho finito la sequenza o se intendo continuare ...

Il massimo di una sequenza immessa da tastiera

Nel computer servono due celle di memoria: una per tenere il "massimo fino a quel momento", e una per mettervi ogni volta il nuovo numero preso dalla tastiera. Chiamiamole **`max`** e **`num`**.

Come dispositivo di input usiamo la tastiera e, solo per comodità in Windows Dev-C++, scegliamo come segnale di fine-sequenza il carattere speciale CTRL/Z, che può venire testato tramite la funzione eof().

Riprendiamo allora l'algoritmo ...

fai l'input di un numero da tastiera
e memorizzalo come massimo
ripeti indefinitamente {
    scrivi un prompt (cioè una richiesta di input)
    esegui l'input da tastiera di un nuovo numero
    se l'input è CTRL/Z esci dal ciclo
    altrimenti {
        se numero > massimo memorizza numero come massimo
    }
}
scrivi massimo

Traduciamolo in C++ una riga per volta

```
double num, max;
```

fai l'input di un numero da tastiera
e memorizzalo come massimo

ripeti indefinitamente {

    scrivi un prompt (cioè una richiesta di input)

    esegui l'input da tastiera di un nuovo numero

    se l'input è CTRL/Z esci dal ciclo

    altrimenti {

        se numero > massimo memorizza numero come massimo

    }

}

scrivi massimo

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
ripeti indefinitamente {
    scrivi un prompt (cioè una richiesta di input)
    esegui l'input da tastiera di un nuovo numero
    se l'input è CTRL/Z esci dal ciclo
    altrimenti {
        se numero > massimo memorizza numero come massimo
    }
}
scrivi massimo
```

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
while(true) {
    scrivi un prompt (cioè una richiesta di input)
    esegui l'input da tastiera di un nuovo numero
    se l'input è CTRL/Z esci dal ciclo
    altrimenti {
        se numero > massimo memorizza numero come massimo
    }
}
scrivi massimo
```

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
while(true) {
   cout << "immetti numero: ";
   cin >> num;
   se l'input è CTRL/Z esci dal ciclo
   altrimenti {
      se numero > massimo memorizza numero come massimo
   }
}
scrivi massimo
```

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
while(true) {
   cout << "immetti numero: ";
   cin >> num;
   if(cin.eof()) break;
   altrimenti {
      se numero > massimo memorizza numero come massimo
   }
}
scrivi massimo
```

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
while(true) {
   cout << "immetti numero: ";
   cin >> num;
   if(cin.eof()) break;
   else {
      if(num > max) max = num;
   }
}
```

scrivi massimo

```
double num, max;
cout << "immetti il primo numero: ";
cin >> max;
while(true) {
  cout << "immetti numero: ";
  cin >> num;
  if(cin.eof()) break;
  else {
    if(num > max) max = num;
  }
}
cout << "il massimo e' " << max << endl;
```

Il massimo di un array

```
double num, max;

max = a[0];
while(true) {
   cout << "immetti numero: ";
   cin >> num;
   if(cin.eof()) break;
   else {
      if(num > max) max = num;
   }
}
cout << "il massimo e' " << max << endl;
```

Il massimo di un array

```
double num, max;

max = a[0];
for(int i = 1; i < n; i++)  {



    if(a[i] > max) max = a[i];

}
cout << "il massimo e' " << max << endl;
```

Una funzione che restituisce il massimo di un array

```
double elementoMax(double a[], int n) {
  double max = a[0];
  for(int i = 1; i < n; i++)
    if(a[i] > max) max = a[i];
  return max;
}
```

Una funzione che restituisce il massimo di un array

Ora vogliamo scrivere una funzione che, dato un array di stringhe, restituisce l'ultima in ordine alfabetico.

È un problema molto diverso?

```
double elementoMax(double a[], int n) {
  double max = a[0];
  for(int i = 1; i < n; i++)
    if(a[i] > max) max = a[i];
  return max;
}
```

Ultima stringa in ordine alfabetico

Ora vogliamo scrivere una funzione che, dato un array di stringhe, restituisce l'ultima in ordine alfabetico.

È un problema molto diverso?

No! è lo stesso problema!

```
string elementoMax(string a[], int n) {
 string max = a[0];
  for(int i = 1; i < n; i++)
    if(a[i] > max) max = a[i];
  return max;
}
```

Schema generico

Si può definire una funzione arrayMax per array di elementi di qualunque tipo sul quale sia definita una relazione d'ordine: basta cambiare il nome del tipo nei tre punti in cui compare.
Lo schema della funzione è dunque, se indichiamo con T un tipo arbitrario:

```
T elementoMax(T a[], int n) {
  T max = a[0];
  for(int i = 1; i < n; i++)
    if(a[i] > max) max = a[i];
  return max;
}
```

I templates

In C++ vi è un costrutto che permette di scrivere la funzione arrayMax una volta sola, per un tipo generico.
Il nome del tipo generico, così come i nomi dei parametri, è arbitrario, ma la convenzione stilistica vuole che sia una lettera maiuscola (ad es. E iniziale di elemento, o T di tipo, ecc.).

```
template <typename T> T elementoMax(T a[], int n) {
  T max = a[0];
  for(int i = 1; i < n; i++)
    if(a[i] > max) max = a[i];
  return max;
}
```

Esercizi

1) Scrivi un template di funzione che, dato un array, restituisca l'indice dell'elemento di valore massimo:

```
template <typename T> int indiceMax(T a[], int n)
```

2) Scrivi un template di funzione che, dato un array, restituisca l'elemento di valore minimo.

3) Scrivi un template di funzione che, dato un array, restituisca l'indice del minimo.