Introduzione alla programmazione in C

Stefano Cherubin*

22/10/2015

[**Informatica A**] Esercitazione #**3**

corso per Ing. Gestionale a.a. 2015/16

Indice

*<nome.cognome>@polimi.it

1 Esercizio: numeri non decrescenti

> Buongiorno ragazzi, benvenuti all'esame di Informatica A. Scrivete un programma in linguaggio C che risolva il seguente problema. Letti tre numeri interi a, b, c dallo standard input, stampare a terminale la sequenza dei tre numeri in ordine non decrescente.

Esempio: a = 10, b = 7, c = 9 deve dare in uscita 7 9 10.

1.1 Approccio al problema

Quando si parla di programmazione, la formulazione di un problema contiene quasi sempre

- vincoli
- descrizione della logica
- consigli su come trovare più facilmente una soluzione al problema
- elementi inutili o fuorvianti[1]
- casi di test o esempi di output atteso

Sta al programmatore riconoscere questi elementi e gestirli al meglio delle sue possibilità.

1.1.1 Analisi del testo dell'esercizio

Buongiorno ragazzi elemento totalmente fuorviante. Oggi dovrete concentrarvi e produrre cose sensate, non ci sarà spazio per il divertimento.

benvenuti all'esame elemento inutile. Si tratta solo di convenevoli.

di informatica A consiglio. Se una volta ricevuto in mano il foglio dell'esame vi siete scordati di cosa si sta parlando, probabilmente dovrete ricordarvi delle lezioni di Informatica A.

Scrivete un programma vincolo. L'esercizio è un test di produzione e verrete valutati sulla capacità di scrivere un programma.

in linguaggio C vincolo. In questo esercizio dovrete rispettare lo standard previsto dal linguaggio C. La capacità di attenersi a questo stanrdard sarà oggetto di valutazione.

che risolva il seguente problema elemento inutile. Dovrebbe essere sottinteso.

Letti tre numeri descrizione della logica. Dovete fare questo.

[1] in sede di esame questi elementi tenderanno ad essere minimizzati. Al di fuori, costituiscono una parte decisamente non trascurabile.

interi vincolo. Dovrete lavorare con numeri interi.

a, b, c consiglio. Nello specifico caso, questo non è un brutto modo di chiamare i dati su cui lavorate.

dallo standard input vincolo. I dati dovranno essere acquisiti dallo standard input.

stampare [...] la sequenza dei tre numeri descrizione della logica.

a terminale vincolo. L'output deve essere emesso sul terminale, quindi standard output.

in ordine non decrescente vincolo. La parte di logica che non è descritta nel testo dovrà essere studiata per soddisfare questo vincolo. L'impostazione di tale logica è lasciata alla libera interpretazione del programmatore.

Esempio [...] Si tratta di un esempio. Al termine della produzione del programma, si consiglia di usare questi dati per eseguire una simulazione di esecuzione del programma e verifcare che l'output fornito in simulazione coincida con l'output atteso.

1.2 Iniziare la stesura

Partendo dai vincoli e dalla logica, si possono identificare alcuni elementi ricorrenti che possono essere tradotti immediatamente in codice.

1.2.1 Scrivere un programma in linguaggio C

```
int main() {

    return 0;

}
```

1.2.2 tre numeri interi

```
int main() {

    int a, b, c;
    return 0;

}
```

1.2.3 Letti tre numeri interi dallo standard input

```
#include <stdio.h>  /* necessaria se voglio utilizzare
                       standard input o standard output */
int main() {
```

```
    int a, b, c;
    printf("\n Inserisci il numero a: ");
    scanf("%d",&a);
    printf("\n Inserisci il numero b: ");
    scanf("%d",&b);
    printf("\n Inserisci il numero c: ");
    scanf("%d",&c);

    return 0;
}
```

1.2.4 Stampare a terminale la sequenza dei numeri

```
#include <stdio.h>
int main() {

    int a, b, c;
    printf("\n Inserisci il numero a: ");
    scanf("%d",&a);
    printf("\n Inserisci il numero b: ");
    scanf("%d",&b);
    printf("\n Inserisci il numero c: ");
    scanf("%d",&c);

    printf("\n L'ordine voluto e': %d, %d, %d",a,b,c);
    return 0;

}
```

1.3 Soluzione 1 - analisi per casi

Un possibile approccio al problema è quello di valutare tutti i possibili ordinamenti dei numeri interi in input e, in base ad opportuni controlli, eseguire l'unica tra le istruzioni di output che utilizza l'ordinamento corretto.

```
#include <stdio.h> /* inclusione della libreria standard */
int main( ) {

  int a,b,c;       /* dichiarazione delle variabili      */

  /* legge tre interi a,b,c dallo standard input */
  printf("\n Inserisci il numero a: ");
  scanf("%d",&a);
  printf("\n Inserisci il numero b: ");
  scanf("%d",&b);
  printf("\n Inserisci il numero c: ");
```

```
  scanf("%d",&c);

  if (a < b) {
    if (b < c) {
      printf("\n L'ordine voluto e': %d, %d, %d",a,b,c);
    }
    else {
       if (a < c) {
         printf("\n L'ordine voluto e': %d, %d, %d",a,c,b);
      }
      else {
         printf("\n L'ordine voluto e': %d, %d, %d",c,a,b);
      }
    }
  }
   else {
    if (c < b) {
       printf("\n L'ordine voluto e': %d, %d, %d",c,b,a);
    }
    else {
       if (a < c) {
         printf("\n L'ordine voluto e': %d, %d, %d",b,a,c);
      }
      else {
         printf("\n L'ordine voluto e': %d, %d, %d",b,c,a);
      }
    }
   }
  return 0;
}
```

1.4 Soluzione 2 - ordinamento dell'input

Un secondo approccio alla risoluzione del problema prevede di fissare un solo dove eseguire l'output ed eseguire delle istruzioni utili a rendere i dati acquisiti coerenti nel loro ordinamento con quanto richiesto in output.

```
#include <stdio.h> /* inclusione della libreria standard     */
int main( ) {


    int a,b,c,t; /* dichiarazione delle variabili              */

    /* legge tre interi a,b,c dallo standard input             */
    printf("\n Inserisci il numero a: ");
    scanf("%d",&a);
```

```
    printf("\n Inserisci il numero b: ");
    scanf("%d",&b);
    printf("\n Inserisci il numero c: ");
    scanf("%d",&c);

    /* ordinamento dei valori delle variabili a,b    */
    if (a > b) {

       /* Scambio dei valori delle due variabili a,b  */
       t = a;
       a = b;
       b = t;
    }

    /* ordinamento dei valori delle variabili a,c    */
    if (a > c) {

       /* Scambio dei valori delle due variabili a,c  */
       t = a;
       a = c;
       c = t;
    }
    /* la variabile a contiene ora sicuramente il    */
    /* valore più piccolo tra quelli inseriti.       */

    /* ordinamento dei valori delle variabili b,c    */
    if (b > c) {

       /* Scambio dei valori delle due variabili b,c  */
       t = b;
       b = c;
       c = t;
    }
    printf("\n L'ordine voluto e': %d, %d, %d",a,b,c);
    return 0;
}
```

2 Esercizio 2

Si scriva un programma in linguaggio C che letto un numero intero positivo dallo standard input, visualizzi a terminale il quadrato del numero stesso facendo uso soltanto di operazioni di somma.

Si osservi che il quadrato di ogni numero intero positivo N può essere costruito sommando tra loro i primi N numeri dispari.

Esempio:

$$\begin{aligned} N &= 5 \\ N^2 &= 1+3+5+7+9=25 \end{aligned}$$

2.1 Approccio alla soluzione

2.1.1 Premessa

L'idea di soluzione è quella di scandire i primi N numeri dispari esprimendoli nella forma $(i+i+1)$ al variare dell'intero $i \in [0; N-1]$ e accumulare la loro somma man mano che si procede nella loro scansione in un'altra variabile.

$$\begin{aligned} N^2 &= \sum_{i=0}^{N-1} 2 \cdot i + 1 \\ &= \sum_{i=0}^{N-1} i + i + 1 \\ &= (2{\cdot}0+1) + \ldots + (2{\cdot}i+1) + \ldots + (2{\cdot}(N-1)+1) \end{aligned}$$

quindi si utilizzeranno almeno 3 variabili:

N numero in input

i contatore di iterazioni

S accumulatore

2.1.2 Pseudocodice (stesura informale dell'algoritmo)

1. Inizio dell'algoritmo
2. Leggi un numero intero positivo dallo standard input.
3. Inizializza un contatore i a 0
4. Inizializza un accumulatore S a 0
5. Finché il valore del contatore è minore del numero letto

(a) Somma all'accumulatore il doppio del valore del contatore incrementato di 1

(b) Incrementa il contatore

(c) Torna al punto 5.

6. Stampa a terminale il valore dell' accumulatore

7. Fine dell' algoritmo

2.1.3 Blindare l'input

Quando viene richiesto che un numero in input rispetti determinate caratteristiche (in questo caso che sia positivo) è possibile scrivere il programma in modo che non accetti dati non conformi alle caratteristiche richieste e continui a chiedere all'utente di fornire dei dati fino a quando l'utente non inserisce dati corretti.

Questo è possibile farlo grazie a un ciclo do-while con controllo sul dato immesso.

```
do {

    printf("\n Inserisci un numero positivo N: ");
    scanf("%d",&N);

} while (N <=0);
```

Variante con messaggio di errore A volte può capitare che debbano essere inseriti più dati dello stesso tipo e il messaggio che invita l'utente a inserire i dati può non variare. In queste situazioni può essere utile segnalare quando si verifica un errore e non è stato accettato il dato.

```
do {

    printf("\nInserisci un numero positivo N: ");
    scanf("%d",&N);
    if (N <=0) {


        printf ("\nErrore: il numero inserito non e' positivo.");
    }

} while (N <=0);
```

Variante con ciclo while

```
printf("\nInserisci un numero positivo N: ");
scanf("%d",&N);
while (N <= 0) {
```

```
    printf ("\nErrore: il numero inserito non e' positivo.");
    printf("\nInserisci un numero positivo N: ");
    scanf("%d",&N);
}
```

2.2 Soluzione C

```
#include <stdio.h> /* inclusione della libreria standard */
int main( ) {
    /* dichiarazione delle variabili */
    int i; /* contatore */
    int N; /* variabile di cui si vuole calcolare il quadrato */
    int S; /* accumulatore per il risultato del calcolo */

    /* ciclo di controllo che garantisce N > 0 */
    do {

       printf("\n Inserisci un numero positivo N: ");
       scanf("%d",&N); /* legge N dallo standard input */
    /* Finché il numero inserito non è positivo ripetere */
    /* l'immissione dati. */
    } while (N < =0 );
    /* Il numero inserito è positivo */

    S = 0; /* inizializzazione della variabile di accumulo */
    /* ciclo di scansione dei primi N numeri dispari */
    i = 0;
    while(i < N) {

       /* Finché il contatore è minore del numero letto */
       /* aggiorna il contenuto della variabile accumulatore */
       S = S + (i+i+1);
       i = i + 1; /* incrementa il contatore */
    }
    printf("\n Il quadrato del numero inserito e': %d \n",S);
    return 0;
}
```

Licenza e crediti

Crediti

Quest'opera contiene elementi tratti da materiale di Gerardo Pelosi redatto per il corso di Fondamenti di Informatica per Ingegneria dell'Automazione a.a. 2013/14.

Licenza beerware[2]

Quest'opera è stata redatta da Stefano Cherubin. Mantenendo questa nota, puoi fare quello che vuoi con quest'opera. Se ci dovessimo incontrare e tu ritenessi che quest'opera lo valga, in cambio puoi offrirmi una birra.

[2] http://people.freebsd.org/~phk/