

Introduzione alla programmazione in C

Stefano Cherubin*

22/10/2015

[Informatica A] Esercitazione #3

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

Indice

1	Esercizio: numeri non decrescenti	2
1.1	Approccio al problema	2
1.1.1	Analisi del testo dell'esercizio	2
1.2	Iniziare la stesura	3
1.2.1	Scrivere un programma in linguaggio C	3
1.2.2	tre numeri interi	3
1.2.3	Letti tre numeri interi dallo standard input	3
1.2.4	Stampare a terminale la sequenza dei numeri	4
1.3	Soluzione 1 - analisi per casi	4
1.4	Soluzione 2 - ordinamento dell'input	5

*<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 fuorvianti¹
- 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 standard sarà oggetto di valutazione.

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

Letti tre numeri descrizione della logica. Dovete fare questo.

¹in 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 verificare 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;
}

```

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 beerware²

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/>