

Esercitazioni di Fondamenti di Informatica - Lez. 3

07/10/2020

Algoritmi di codifica

A- individuare i dati di input e di output, che diventeranno variabili del programma

B- descrivere la soluzione utilizzando le strutture di controllo fondamentali e descriverlo in linguaggio naturale o pseudocodice

D- tradurlo in linguaggio C scegliendo opportunamente le istruzioni

```
// calcolo del fattoriale
//
// leggere
// verifico N>0
// se N=0 fatt = 1
// se N >1    calcolo il fattoriale
//    conto per c da 1 a N
//    fatt = fatt * c
// printf(fatt)
```

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

int N, fatt, c;

printf("inserire un numero intero ..\n");
do{
    scanf("%d",&N);
}while(N<=0);

printf("leggo %d \n", N);

fatt= 1; // minimo valore calcolabile
if(N>0){
    for(c=1; c<=N; c++)
        fatt = fatt*c;
}
printf("il fattoriale di %d è %d \n",N,fatt);
return 0;
}
```

```
// media dei numeri dispari di N inseriti
// terminati con 0

// idea:
// leggo numeri finché carattere terminatore 0
// se il numero è dispari sommo e aumento i
```

```
// leggo N
// se resto N%2 > 0
// somma = somma+N
// i= i+1
// finche N!=0
// stampa media = somma/i
```

```
#include <stdio.h>
int main() {
    int N, i, somma ;

    scanf("%d",&N);
    i=0;
    somma = 0;

    while(N!=0){
        if (N%2>0){
            somma= somma+N;
            i= i+1;
        }
        scanf("%d",&N);
    }

    printf("media dispari %d", somma/i);
    return 0;
}
```

```
// dato un numero N e una base b: convertire N in base b
// con N decimale e base <10
//
//
// leggere N e base
// verificare che siano positivi con base <10
// finché N > 0
// calcolo il resto col modulo e lo memorizzo in array r[i]
// Eseguo la divisione N= N/b
// fine finché
// stampo r[c] dal indice massimo a 0 poiché i resti sono memorizzati per primi
```

```

#include <stdio.h>
int main() {
    int N, b, resti[15], i,j;

    // nota con 16 bit memorizzo circa 32000 int decimali

    printf("inserire N e una base di conversione minore di 10 \n");
    do{
        printf("inserire numero \n");
        scanf("%d", &N);
    }while( N<=0);
    int t=N;
    do{
        printf("inserire base\n");
        scanf("%d", &b);
    }while(b>10);

    i=0;
    while(N>0){
        resti[i]= N%b;
        N=N/b;
        i=i+1;
    }
    printf("conversione in base  %d di %d \n", b,t);
    for(j=i-1; j>=0; j--){
        printf("%d", resti[j]);
    }

    return 0;
}

```

// esercizio fatto da Baresi

//stampare i divisori di un numero N

// osservo che un numero D è divisore se il modulo è 0

// parto da D = 2

// quando ho trovato un divisore, divido ancora per lo stesso D

// finché il resto è 0

// poi D=D+1

// finché N vale 1

```

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

    do{

```

```

printf("inserire numero \n");
scanf("%d", &N);
}while( N<=0);

```

d= 2; // comincio da 2 perché 1 è sempre divisore

```

while( N>1) {

    while( N%d = 0 ) {
        printf("%d \n", d);
        N = N/d;
    }
    d = d+1;
} // fine while
}

```

//esercizio fatto da Baresi
 // somma dei primi n numeri (ricordando Gauss)

```

#include <stdio.h>

int main() {

    int numero, i;
    int somma=0;
    printf("inserisci un numero: ");
    scanf("%d", &numero);
    for(i=0; i<numero; i++) {
        somma+=i;
    }
    printf("la somma e' %d:\n ", somma);

    return 0;
}

```

Si verifica l'osservazione di Gauss per cui la somma degli estremi fornisce sempre il valore di N

0 1 2 3 4 5 6 7 8 9 10 11 12

0+11 1+10 3+9 il numero delle coppie è N/2

Nei numeri pari il valore centrale rimane da solo e corrisponde a N/2

Risultato si ottiene anche $N \cdot N/2$ con N dispari
 $N \cdot N/2 + N/2$ con N pari

Esercizi di codifica in C

I codici e le soluzioni degli esercizi sono nella cartella **codice1**

1. Dato il seguente codice, trovare gli errori, correggerli e indicare cosa si ottiene come output.

```
0 #include <stdio .h>
1 int main (){
2 int i;
3 int j;
4 scanf ("%d%d",i,j);
5 if(i<j) ;
6 printf (" Minore !");
7 }
```

Soluzione:

```
#include <stdio .h>
int main (){
int i;
int j;
scanf ("%d%d",&i, &j); // "all'indirizzo di ...."
if(i<j) { // ; termina il comando if quindi
// l'istruzione seguente veniva eseguita in ogni caso
// aggiungere le graffe serve a indicare le istruzioni o righe
// da eseguire dopo if
// NB non viene segnalato dal compilatore è un errore
semantico
printf (" Minore !");
}
} // fine main
```

Cosa fa il programma:

Il programma legge due numeri interi e stampa la stringa "Minore!" se il primo numero è minore.

Non stampa niente in caso contrario.

2. Il seguente codice dovrebbe: leggere due numeri e comunicare in output se sono uguali o diversi. Trovare l'errore e correggerlo.

```
#include <stdio.h>
void main (){
    int i;
    int j;
    scanf ("%d%d",i,j);
    if(i=j)
        printf ("%d e %d sono uguali !",i,j);
    else {
        printf ("%d e %d sono diversi !",i,j);
    }
}
```

Soluzione:

```
#include <stdio.h>
int main(){
    int i;
    int j;
    scanf ("%d%d",&i,&j); //mancava & per scrivere all'indirizzo
                          //della variabile
    if(i==j){            // l'operatore logico è diverso da uguaglianza
        printf ("%d e %d sono uguali !",i,j);
    }
    else {
        printf ("%d e %d sono diversi !",i,j);
    }
    return 0;
}
```

3. Spiegare cosa calcola il seguente codice.

```
0 #include <stdio.h>
1 int main(){
2     int i,j=1;
3     scanf("%d",&i);
4     while (i >0){
5         j=j*i;
6         i=i-1;
7     }
8     printf ("%d\n",j);
9     return 0;
10 }
```

Soluzione:

Per capire cosa calcola il codice è possibile fare una **simulazione della sua esecuzione** supponendo un valore di input per la variabile i e costruendo una tabella con i valori delle variabili dopo ogni esecuzione.

tabella di traccia

Istruzione	i	j	Commento
int i,j=1;	non iniz.	1	dichiaro variabili, inizializzo j a 1, i non ha ancora valore
scanf("%d",&i);	4	1	leggo da input il valore ipotizzato 4, salvato in i
while(i>0)	4	1	controllo se 4 > 0 ed eseguo le operazioni seguenti
j=j*i;	4	4	eseguo 14 e lo assegno a j
i=i-1;	3	4	decremento di 1 il valore di i
while(i>0)	3	4	controllo se 3 > 0, il ciclo continua
j=j*i;	3	12	eseguo 43 e lo assegno a j
i=i-1;	2	12	decremento di 1 il valore di i
while(i>0)	2	12	controllo se 2 > 0, il ciclo continua
j=j*i;	2	24	eseguo 122 e lo assegno a j
i=i-1;	1	24	decremento di 1 il valore di i
while(i>0)	1	24	controllo se 2 > 0, il ciclo continua
j=j*i;	1	24	eseguo 241 e lo assegno a j
i=i-1;	0	24	decremento di 1 il valore di i
while(i>0)	0	24	controllo se 0 > 0, il ciclo si ferma
printf("%d ",j)	0	24	stampo il valore di j

Posso notare come inserendo il valore 4 per la variabile i, ho eseguito la moltiplicazione $4 \cdot 3 \cdot 2 \cdot 1 = 24$, che è il fattoriale di 4 (scritto anche $4!$).

Il codice quindi calcola il fattoriale di un numero dato in input.

4. Dato il seguente codice, indicare cosa si ottiene come output quando $a = 5.0$ e $b = 2.0$.

	tabella di traccia		
	a	b	c
0 #include <stdio.h>			
1 int main(){			
2 float a,b;			
3 int c;			
4 scanf ("%f%f" ,&a ,&b);	5.0	2.0	
5 c=a/b;	5.0	2.0	2
6 printf ("%f",c);	5.0	2.0	errore di formato
7 }			

Soluzione:

Supponendo di leggere $a = 5.0$ e $b = 2.0$, la loro divisione avrebbe risultato 2.5. Assegnando tale valore alla variabile intera c, la parte decimale viene persa e abbiamo che $c = 2$. La stampa però non è corretta, perché si tenta di stampare un numero intero come se fosse un numero a virgola mobile ("%f"). La funzione printf non effettua la conversione tra tipi di dato con rappresentazioni differenti!

Se in riga 6 scrivessimo `printf("%d",c);`, il codice stamperebbe 2. La versione corretta del codice si ottiene dichiarando c come un float, invece che un int.

Versione corretta:

```
#include <stdio.h>
int main(){
    float a,b;
    float c;           // tipo corretto
    scanf ("%f%f" ,&a ,&b);
    c=a/b;
    printf ("%f",c);
}
```


5. Qual é il contenuto dell'array *a* al termine dell'esecuzione del seguente codice?

```
# include <stdio.h>
# define MAX 100
int main (){
    int a[MAX] = {1, 2, 3, 4, 5};
        a[1]= a[2];
        a[a[1]] = a[0];
        a[5]= a[0];

    return 0; }
```

Soluzione:

Come per l'esercizio precedente, possiamo **simulare l'esecuzione dell'algoritmo** passo per passo, in questo caso controllando il valore dell'array *a* dopo ogni istruzione e il valore dell'indice usato per modificare un valore di *a*. Ricordiamo che in C gli indici degli array incominciano dal valore 0.

Istruzione	a	Commento
int a[<i>MAX</i>] = 1, 2, 3, 4, 5;	[1, 2, 3, 4, 5]	inizializzo <i>a</i> di lunghezza 100 con i valori da 1 a 5
<i>a</i> [1] = <i>a</i> [2];	[1, 3, 3, 4, 5]	sostituisco il secondo valore (indice 1) con il terzo (indice 2)
<i>a</i> [<i>a</i> [1]] = <i>a</i> [0];	[1, 3, 3, 1, 5]	<i>a</i> [1] vale 3, quindi sostituisco il quarto valore con il primo
<i>a</i> [5] = <i>a</i> [0];	[1, 3, 3, 1, 5, 1]	<i>a</i> [5] non era specificato, ma esiste, lo inizializzo con il primo

6. Il seguente codice dovrebbe calcolare il massimo di un array letto in input.

Trovare gli errori e correggerli.

```
1 #include<stdio.h>
2 #include<limits.h>
3 // limits.h contiene le definizioni di INT_MIN e INT_MAX
4 // cioè il numero intero minore e maggiore rappresentabile in C
5 #define MAX -5
6 int main(){
7     int a[MAX],i=0, massimo=INT_MIN;
8     while(i<=MAX){
9         scanf("%d",&a[i]);
10    while(i>0){
```

```

11  if(massimo> a[i])
12      massimo = i;
13  i=i-1;
14  }
15  printf("%d\n",massimo)
16  }

```

Soluzione:

legge un array da input (per es. `a[] = [4 3 66 32 0]`), inizializza *massimo*= -2147483648 ,
 confronta ogni elemento con *massimo* e se maggiore, lo sostituisce poi stampa *massimo*
 note: gli errori sono nelle righe 5,9/10,11, 12

```

#include <stdio.h>
1 // limits.h contiene le definizioni di INT_MIN e INT_MAX cioe'
2 // il numero intero minore e maggiore rappresentabile in C
3 #include <limits.h>
4 // N.B. definire MAX con un numero negativo causa un errore in
5 // compilazione , perche' viene usato per definire la lunghezza
6 // di un array e non si puo' definire un array con lunghezza
7 // negativa
8 #define MAX 5
9 int main(){
10 int a[MAX],i=0,massimo=INT_MIN;
11 //N.B. la condizione non puo' essere i<=MAX, ma i < MAX
12 // altrimenti si accedrebbe ad a[MAX] che non e' definito
13 while(i<MAX){
14 scanf("%d",&a[i]);
15 // manca un'istruzione per modificare il contatore
16 // il ciclo continuerebbe all'infinito
17 i= i + 1
18 // manca la parentesi di chiusura del ciclo
19 }
20 while(i>0){
21 // alla prima iterazione di questo ciclo , i == MAX
22 // devo quindi decrementarlo di 1
23 // prima di poter accedere all'array
24 i = i - 1;
25 // la condizione di controllo e' sbagliata
26 // aggiorno il massimo se e' minore del valore di a
27 // controllato
28 if(massimo < a[i])
29 // devo aggiornare il massimo con il valore a[i]
30 // non con l'indice
31 massimo = a[i];
32 }
33 printf("%d\n",massimo);
34 }

```

7. Spiegare cosa produce il seguente codice, si consiglia di aiutarsi con una tabella di traccia.

```
0 # include <stdio.h>
1 # define MAX 10
2 int main () {
3     char a[ MAX] = {'c','i','a','o',' ',' '};
4     int i = 4;
5     while (i<MAX ) {
6         scanf ("%c" ,&a[i]);
7         i=i+1;
8     }
9     i=0;
10    while (i<MAX ) {
11        printf ("%c",a[i]);
12        i=i+1;
13    }
14 }
```

Soluzione:

righe 5-8 effettua un primo ciclo di lettura in cui legge dei caratteri fino a riempire l'array a.

righe 4-5 la lettura comincia dalla posizione 4 dell'array, quindi sovrascriverà il carattere spazio ' '.

Il ciclo legge (con scanf) quindi 6 caratteri prima di terminare, dalla posizione di partenza 4 (compresa) all'ultima posizione dell'array 9.

righe 8-9 Terminato il ciclo di lettura avremo quindi un array di caratteri contenente 'ciao' nelle posizioni 0-3 a cui si concatenano i caratteri letti nelle posizioni 4-9

riga 9 Terminata la lettura

righe 10-13 , l'algoritmo stampa tutti i caratteri dell'array.

Supponendo di leggere come input questa sequenza di 6 caratteri 'mondo!', l'array a conterrà 'ciaomondo!', che è anche la sequenza di caratteri stampata.

Codifica i seguenti algoritmi in linguaggio C

I codici e le soluzioni degli esercizi sono nella cartella **codice2**

8. Leggere un array di interi terminato da 0 e stampare la differenza tra il valore massimo e il minimo letti.

```
0  # include <stdio.h>
1  int main (){
2  // non mi serve veramente un array se devo solo calcolare
3  // la differenza tra massimo e minimo !
4
5  int valore ; // intero letto
6  int i = 0;    // contatore valori letti
7  int massimo , minimo ;
8  // inizializzo massimo e minimo al primo valore letto
9  massimo = minimo = valore ;
10 // continuo a leggere fino a che non viene letto 0
11 while ( valore !=0){
12     if( valore > massimo ){
13         massimo = valore ;
14     }
15     if(valore < minimo ){
16         minimo = valore ;
17     }
18     // leggi il prossimo valore
19     scanf ("%d" ,& valore );
20 }
21 int differenza = massimo - minimo ;
22 printf ("max %d - min %d = %d",massimo ,minimo ,
differenza );
23 }
```

con array

```
0  # include <stdio.h>
1  int main (){
2  // leggo array
3  // lo scorro e scelgo max e min
4  int a[50];
5  int valore ; // intero letto
6  int i = 0, dim ;    // contatore valori letti
7  int massimo=0 , minimo=0 ;
8  // inizializzo massimo e minimo al primo valore letto
9  // continuo a leggere fino a che non viene letto 0
10 while ( valore !=0 ){
11     scanf ("%d" ,&valore );
12     a[i]=valore;
13     i=i+1;
14 }
15 dim= i;
16 for (i=0; i<dim; i++) {
17     if (massimo<a[i])
18     {
19         massimo=a[i];
20     }
21     if (minimo>a[i])
22     {
23         minimo=a[i];
24     }
25 }
```

```

24         minimo=a[i];
25     }
26 }
27 int differenza = massimo - minimo ;
28 printf ("max %d - min %d = %d",massimo ,minimo ,
differenza );
29 }

```

9. Scrivere due numeri interi positivi (>0) in input ed effettuare il loro prodotto usando solo l'operatore somma.

```

0 #include <stdio .h>
1 int main (){
2     int a;
3     int b;
4     scanf ("%d %d" ,&a ,&b);
5     // A*B = somma di A per B volte
6     // uso B come contatore
7     // uso una variabile per tenere la somma corrente
8     int prodotto = 0;
9     while (b >0){
10    // aggiornno la somma
11        prodotto = prodotto +a;
12    // decremento il contatore
13        b = b -1;
14    }
15    printf ("Il prodotto e' = %d\n",prodotto );
16 }

```

10. Leggere due numeri interi positivi (> 0) A e B e visualizzare in ordine decrescente la sequenza di numeri interi tra A e B (compresi) che sono divisibili per il minore tra A e B.

Ad esempio, se A=7 e B = 35, la sequenza è 35,28,21,14,7

Provare a risolvere in autonomia per la prossima lezione.