

Fondamenti di Informatica L-A

Esercitazione 4

Programmazione strutturata: blocchi, condizioni, cicli.
Istruzioni switch, do...while, while, for.

Array.

Paolo Torroni

Università degli Studi di Bologna
Laurea in Ingegneria Elettronica e delle Telecomunicazioni

Anno Accademico 2007/2008

△ 1. Istruzione condizionale switch ▽

- ▶ Si consideri il seguente programma C:

```
#include <stdio.h>
main() {
    char A1; int N=10;
    printf( "Please type your command (up/down)...\t" );
    scanf( "%c", &A1 );
    if( A1=='u' ) N++;
    else if( A1=='d' ) N--;
    printf( "N is now %d\n", N );
}
```

- ▶ Si modifichi il codice nel modo seguente:
 - ▶ si usi l'istruzione `switch` al posto degli `if` in cascata.
 - ▶ il programma non deve tener conto delle maiuscole/minuscole nell'input dell'utente (cioè funzioni con 'u', 'U', 'd', e 'D').
 - ▶ l'utente deve essere in grado di specificare N da input.
 - ▶ deve essere presente un ulteriore comando, 'm' (oppure 'M') che
 1. legga da input un altro numero, M;
 2. aggiorni N con il valore $N \times M$.

△ 2. Cicli do...while ▽

- ▶ Si consideri il seguente programma C:

```
#include <stdio.h>
main() {
    char A1=0; int N=10;
    do {
        printf( "Please type a command (up/down)...\t" );
        fflush( stdin ); scanf( "%c", &A1 );
    } while ( A1 != 'u' && A1 != 'd');
    if( A1=='u' ) N++; else if( A1=='d' ) N--;
    printf( "N is now %d\n", N );
}
```

- ▶ Si modifichi il codice in modo che i comandi vengano eseguiti non su un intero N, ma su un array di $m \leq 10$ interi:
 1. Inizialmente, il programma deve chiedere all'utente quanti interi inserire ($0 \leq m \leq 10$).
 2. Poi, il programma deve leggere gli m interi.
 3. Una volta terminata la lettura, il programma deve chiedere il comando (u/d) ed eseguirlo su tutti gli interi
 4. Infine, il programma deve stampare tutti i valori modificati.

△ 3. Cicli while ▽

- ▶ Si consideri il seguente programma C:

```
#include <stdio.h>
main() {
    int N=18, M=15;
    printf( "mcm(%d,%d): ", N, M );
    while ( N!=M )
        if( N>M ) N -= M;
        else M -= N;
    printf( "%d\n", N );
}
```

- ▶ Si apportino le seguenti modifiche al programma:
 1. Si faccia in modo che il programma iteri l'esecuzione con 10 coppie di valori diverse (M,N).
 2. Per ciascuna coppia di valori, il programma tenga traccia del numero K di operazioni di **confronto** ($N!=M$ e $N>M$) effettuate, e del risultato R ottenuto.
 3. Al termine del ciclo, il programma stampi, per ciascuna delle 10 coppie, i valori M , N , K , R . (NOTA: *Bisogna tener traccia dei valori originali di M , N !*)

△ 4. Cicli for ▽

- ▶ Si consideri il seguente programma C:

```
#include <stdio.h>
main() {
    int N=18, M=15, i, A=1;
    printf( "mcm(%d,%d): ", N, M );
    for( i=2; i<N; i++ )
        if( !(N%i) && !(M%i) )
            A=i;
    printf( "%d\n", A );
}
```

- ▶ Si apportino le modifiche specificate al punto 3, con una differenza nel seguente punto:
 2. Per ciascuna coppia di valori, il programma tenga traccia del numero K di operazioni **modulo** ($N\%i$ e $M\%i$) effettuate, e del risultato R ottenuto.