CORSO DI PROGRAMMAZIONE
A.A. 2015-16

Dispensa 8

Laboratorio

Dott. Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it

Istruzione while

I cicli permettono di ripetere delle operazioni fino a quando una condizione risulta vera. Consideriamo la sintassi del ciclo *while*:

```
while (espressione)
{
      Istruzione
}
```

l'espressione viene valutata, qualora sia diversa da zero, *istruzione* è eseguita ad *espressione* rivalutata. Questo ciclo prosegue finché *espressione* diventa zero, momento in cui l'esecuzione riprende dal punto seguente a *istruzione*.

Consideriamo il seguente esempio:

```
A
while (espressione)
{
      B
}
C
```

Dove A, B e C sono blocchi di codice. Viene eseguita l'istruzione A, valutata l'*espressione* e se risulta vera (diversa da zero), si accede al corpo del ciclo *while* eseguendo l'istruzione B, si procede rivalutando l'*espressione*, se risulta vera si procede nuovamente eseguento l'istruzione B, diversamente si esegue l'istruzione C.

Per stampare 7 volte il nome "Cesena" si potrebbe eseguire la seguente porzione di codice:

```
i = 0;
while (i < 7)
{
      printf("Cesena\n");
      i++;     //equivale a fare i = i + 1;
}
```

Consideriamo ora la sintassi del ciclo for:

```
for (espressione1; espressione2; espressione3)
{
      istruzione;
}
```

È equivalente a:

```
espressione1;
while (espressione2)
{
      istruzione;
      espressione3;
}
```

Ad eccezione del comportamento dell'istruzione *continue*, descritto successivamente.

Sul piano grammaticale le tre componenti di un ciclo *for* sono espressioni. Nel caso più comune *espressione1* e *espressione3* sono assegnamenti o chiamate di funzioni, mentre espressione2 è un espressione relazionale. Si può omettere una delle tre, ma i punti e virgola devono rimanere. Omettere *espressione1* o *espressione3* nell'istruzione *for* corrisponde a ometterle nell'equivalente ciclo *while*. Se però la condizione, cioè *espressione2*, non è presente, esse è considerata sempre vera:

```
for (;;)
{
      istruzione;
}
```

Il ciclo è "infinito", presumibilmente da interrompere, prima o poi, con altri strumenti come *break* o *return*. Se sia meglio usare *while* o *for* è sostanzialmente una questione di preferenza personale.

Per stampare 7 volte il nome "Cesena" si potrebbe eseguire la seguente porzione di codice:

```
for (i = 0; i < 7; i++)
{
      printf("Cesena\n");
}
```

Istruzione do-while

Come visto nelle parti precedenti, i cicli *while* e *for* valutano la condizione di terminazione fin dall'inizio. Il terzo ciclo del C, l'istruzione *do-while*, esamina invece la condizione in coda, dopo ogni iterazione: il corpo è quindi sempre eseguito almeno una volta.

La sintassi del costrutto è:

```
do
{
      istruzione;
}
while (espressione);
```

Si esegue dapprima *istruzione*, e si valuta poi *espressione*: se è vera, si torna a eseguire *istruzione*, e così via. Il ciclo termina quando *espressione* diventa falsa.