

ESERCITAZIONE 13 – (Scaricala da: www.ascuoladi.eu) (ciclo do-while)

Il ciclo a condizione finale (do-while) nel linguaggio C.

Esso ha la condizione posta alla fine del ciclo e fa ripetere il ciclo finché la condizione inserita è vera¹ e quindi si esce dal ciclo quando la condizione è falsa. La sua struttura nel flowchart è la seguente:



Esso si può usare: **1) nei casi in cui è preferibile usare il ciclo for** (cioè quando la condizione di uscita è prestabilita dal programmatore (ad es. nel caso si debba leggere N numeri) **2) nei casi in cui si usa il ciclo while** ovvero quando è l'utente a stabilire la condizione di uscita dal ciclo (ad es. leggi una serie di numeri positivi, finché l'utente non inserisce il numero -1).

La differenza con il ciclo while è che poiché ha la condizione alla fine, tutte le istruzioni poste prima della condizione, saranno eseguite almeno una volta e cioè anche quando la condizione finale è falsa. Pertanto **conviene usare il ciclo do-while**, quando tra la parola DO e la condizione while ci sono poche istruzioni, come ad es. accade nei cicli usati **per controllare la bontà dei dati di ingresso** letti dall'utente. Noi useremo tale ciclo solo per questo ultimo caso.

Nel linguaggio C, esso si codifica così: `do { istruzioni.... } while (condizione);`

Problema1:

Scrivere un algoritmo che letti N numeri (Num) positivi, determini quanti sono i numeri pari (Cpar) inseriti e la somma dei numeri positivi (SPos). Osservazione controllare che N sia ≥ 5 e che i numeri siano positivi. (suggerimento: usare tre cicli con 2 nidificati)

```
# include <stdio.h> /* direttiva che permette di usare le funzioni printf e scanf */
# include <stdlib.h> /* direttiva che permette di usare la funzione system() */
int main()
{ /* dichiarazione delle variabili usate */
  int num,n,i,Cpar,SPos,q,r;
  /* num: e' il numero corrente da leggere; cpar: è il contatore di numeri pari; n: è la
   dimensione scelta dall'utente dei numeri da leggere; i: è il contatore del ciclo For
   SPos è la somma dei numeri positivi; q è il quoziente di num/2 e r il suo resto.
  */
  printf(" CALCOLO DEI PARI E LA SOMMA DEI POSITIVI DI N NUMERI (N>=5)\n");
  /* lettura dati di ingresso N (dimensione) e Numero corrente del programma */
  do { printf("\nN = (Attenzione N>=5)"); scanf("%d",&n); } while(n<5);
  Cpar=0; /* azzeriamo il contatore Cpari dei pari*/
  for(i=0;i<n;i++)
  { /* inizio corpo del programma */
    do { printf("\nNum = (Attenzione Num>0)"); scanf("%d",&num); } while(num<=0);
    SPos=SPos+num; q=num/2; r=num-2*q; if(r==0) Cpar++;
  } /* chiusura del ciclo For */
  printf("\n\nLA SOMMA DEI NUMERI POSITIVI INSERITI E' = %d\n\n",SPos);
  printf("LA QUANTITA' DEI NUMERI PARI E' = %d\n",Cpar);

  system("PAUSE"); /* essa permette di mantenere aperta la finestra Dos */
} /*chiusura del programma principale */
```

By AScuolaDi.eu

¹ Negli altri linguaggi in genere nel ciclo do-while, il ciclo si ripete finché la condizione è falsa.

TEST PROVA DEL PROGRAMMA CREATO:

CONSEGNA LAVORO	CALCOLO DEI PARI E LA SOMMA DEI POSITIVI DI N NUMERI (N>=5 E NUM>0)
SCRITTO SUL QUADERNO	N = <Attenzione N>=5> 2
	N = <Attenzione N>=5> -3
	N = <Attenzione N>=5> 5
1) La tabella delle variabili;	Num = <Attenzione Num>0> -1
	Num = <Attenzione Num>0> 2
2) Il diagramma a blocchi;	Num = <Attenzione Num>0> 4
	Num = <Attenzione Num>0> 1
3) Il test sul diagramma;	Num = <Attenzione Num>0> 1
	Num = <Attenzione Num>0> 1
4) il codice C	LA SOMMA DEI NUMMERI POSITIVI INSERITI E' = 9 LA QUANTITA' DEI NUMERI PARI E' = 2

PROBLEMA 2

Scrivere un algoritmo che letti D numeri (N) negativi, determini quanti sono i numeri negativi (CNeg) inseriti e la somma dei dispari (SDisp). Osservazione controllare che D sia <= 20 e che i numeri N siano negativi. (suggerimento: usare tre cicli con 2 nidificati)

```
#include <stdio.h> /* direttiva che permette di usare le funzioni printf e scanf */
#include <stdlib.h> /* direttiva che permette di usare la funzione system() */
int main() {
    /* dichiarazione delle variabili usate */
    int d,n,i,CNeg,Sdisp,q,r;
    /* n: è il numero corrente da leggere negativo; CNeg: è il contatore di numeri negativi
       d: è la dimensione scelta dall'utente dei numeri da leggere; i: è il contatore del ciclo
       for, Sdisp è la somma dei numeri dispari, q il quoziente, r il resto di n con 2. */
    printf("CALCOLO DEI NEGATIVI E LA SOMMA DEI DISPARI DI D NUMERI (D<=20,D>0, N<0)\n");
    /* lettura dati di ingresso N (numero) e D (dimensione) del programma */
    do {
        printf("\nD = (Attenzione D<=20 e D>0)"); scanf("%d",&d);
    } while(d>20 OR d<=0); /* OR operatore logico di disgiunzione */
    CNeg=0; Sdisp=0; /* azzeriamo il contatore CNeg e la somma dei dispari */
    /* inizio corpo de programma */
    for(i=0;i<d;i++)
    {
        do { printf("\nN = (Attenzione n<0)"); scanf("%d",&n); } while(n>=0);
        CNeg++; q=n/2; r=n-2*q;
        if(r!=0) Sdisp=Sdisp+n;
    } /* chiusura del ciclo For */
    printf("\n\nLA SOMMA DEI NUMMERI DISPARI INSERITI E' = %d\n\n",Sdisp);
    printf("LA QUANTITA' DEI NUMERI NEGATIVI E' = %d\n",CNeg);

    system("PAUSE"); /* essa permette di mantenere aperta la finestra Dos */
} /*chiusura del programma principale */
```

BY AScuolaDi.eu

TEST PROVA DEL PROGRAMMA CREATO:

<p>CONSEGNA LAVORO</p> <p>SCRITTO SUL QUADERNO</p> <p>1) La tabella delle variabili;</p> <p>2) Il diagramma a blocchi;</p> <p>3) Il test sul diagramma;</p> <p>4) il codice C</p>	<p>CALCOLO DEI NEGATIVI E LA SOMMA DEI DISPARI DI D NUMERI <D<=20,D>0, N<0></p> <p>D = <Attenzione D<=20 e D>0> 0</p> <p>D = <Attenzione D<=20 e D>0> 21</p> <p>D = <Attenzione D<=20 e D>0> 4</p> <p>N = <Attenzione n<0> 1</p> <p>N = <Attenzione n<0> 2</p> <p>N = <Attenzione n<0> 3</p> <p>N = <Attenzione n<0> 4</p> <p>N = <Attenzione n<0> 5</p> <p>N = <Attenzione n<0> -3</p> <p>N = <Attenzione n<0> -1</p> <p>N = <Attenzione n<0> -5</p> <p>N = <Attenzione n<0> -2</p> <p>LA SOMMA DEI NUMERI DISPARI INSERITI E' = -9</p> <p>LA QUANTITA' DEI NUMERI NEGATIVI E' = 4</p> <div data-bbox="917 380 1216 546" style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>I primi 5 numeri inseriti sono tutti positivi e perciò il programma non li considera, poichè si aspetta solo numeri negativi.</p> </div>
---	---