

Norme di Leggibilità nel linguaggio



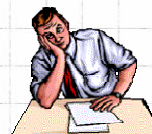
Prof. Luigi Lamberti
2005

Come NON programmare

```
#define Y return
#define R for
#define e while
#define I printf
#define l int
#define W if
#define C y=v+111;H(x,v)*y++= *x
#define H(a,b) R(a=b+11;a<b+89;a++)
#define s(a) t=scanf("%d",&a)
#define U Z I
#define Z I("123\
45678\n");H(x,v){putchar(".XO"[*x]);W((x-v)%10==8){x+=2;I("%d\n", (x-v)/10-1);}}
l V[1600],u,r[]={-1,-11,-10,-9,1,11,10,9},h[l]={11,18,81,88},ih[l]={22,27,72,77},
bz,lv=60,*x,*y,m,t;S(d,v,f,_,a,b)l*v;{l c=0,*n=v+100,j=d<u-1?a:-9000,w,z,i,g,q=
3-f;W(d>u){R(w=i=0;i<4;i++)w+=(m=v[h[i]])==f?300:m=q?-300:(t=v[ih[i]])==f?-50:
t==q?50:0;Y w;}H(z,0){W(E(v,z,f,100)){c++;w= -S(d+1,n,q,0,-b,-j);W(w>j){g=bz=z;
j=w;W(w>=b|w>=8003)Y w;}}W(!c){g=0;W(_){H(x,v)c+= *x==f?1:*x==3-f?-1:0;Y c>0?
8000+c:c-8000;}C;j= -S(d+1,n,q,1,-b,-j);bz=g;Y d>=u-1?j+(c<<3):j;}
main()
{R(;t<1600;t+=100)R(m=0;m<100;m++)V[t+m]=m<11||m>88|| (m+1)%10<2?3:0;I("Level:");
V[44]=V[55]=1;V[45]=V[54]=2;s(u);e(lv>0){Z do{I("You:");s(m);e(!E(V,m,2,0)&&m!=99);
W(m!=99)lv--;W(lv<15&&u<10)u+=2;U("Wait\n");I("Value:&d\n",S(0,V,1,0,-9000,9000
));I("move: %d\n", (lv-=E(V,bz,1,0),bz));}}E(v,z,f,o)l*v;{1*j,q=3-f,g=0,i,w,*k=v
+z;W(*k==0)R(i=7;i>=0;i--){j=k+(w=r[i]);e(*j==q)j+=w;W(*j==f&&j-w!=k){W(!g){g=1
;C;}e(j!=k)*((j-=w)+o)=f;}Y g;}
```



Commenti



La quantità di commenti deve superare abbondantemente il codice C.

Un altro programmatore deve capire come funziona una routine senza chiedervi nulla.

I commenti devono essere scritti PRIMA, in modo da formare l'ossatura su cui verrà scritto il programma, non DOPO!

Nomi di variabili e funzioni

Il Nome deve richiamare l'uso della variabile e/o della funzione.

C è case-sensitive; usare l'iniziale maiuscola per i nomi, per distinguerle a colpo d'occhio dalle parole del C:



```
int      NumLati;  
float    AreaCerchio, Raggio,  
         MaxValore;
```

Definizione di costanti



Utilizzare il **#define** per i valori costanti.

Le costanti vanno scritte tutto in **MAIUSCOLO**.

```
#define PIGRECO 3.1415927
```

```
#define ESC 27
```

Spaziatura



Inserire qualche rigo vuoto tra le diverse funzioni e tra le varie sezioni del programma.

Inserire gli spazi tra i nomi e gli operatori

```
for(i=0,j=N;i<j&& j>=10;i++,j-=3)if(A[i][j])A[i][j]*=i;
```

diventa

```
for ( i = 0, j = N; i < j && j >= 10; i ++, j -= 3)
```

```
    if ( A[i][j] ) A[i][j] *= i;
```

Disposizione Sottoprogrammi



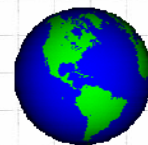
In **C** è usuale inserire le funzioni DOPO il **main**, utilizzando il prototype per la visibilità.

Costruire un **progetto** suddividendo il codice in più **file.c**, uno per ogni gruppo di funzioni.

I **prototype** verranno inseriti in altrettanti **file.h**


Nei file.h saranno inserite anche le definizioni di costanti e i **typedef**.

Variabili Globali




L'uso delle variabili globali è fortemente sconsigliato perché:

- ❖ consente a tutte le procedure di accedere ai dati, con conseguente difficoltà di debug di eventuali errori;
- ❖ rende difficoltoso il riutilizzo del codice;
- ❖ quasi sempre, impedisce l'uso della ricorsione: ogni istanza di una funzione ricorsiva deve agire su un'immagine diversa dei dati.



Uscita da una funzione




Nella programmazione strutturata un blocco di istruzioni deve avere **UN** punto di inizio e **UN** punto di uscita.

Questo vale anche per i sottoprogrammi e il programma principale.

- ❖ Non usare **break** per interrompere un ciclo (salvo al termine di un **case**).
- ❖ L'unico **return** di una funzione deve essere l'ultima istruzione.

Allineamento e Indentazione



Allineare le istruzioni, disponendo opportunamente quelle che sono subordinate ad una condizione o ad un ciclo, qualche colonna più a destra.

I simboli di BEGIN e END, ovvero { e } devono essere sempre allineati con l'istruzione cui fa riferimento il blocco (**if, else, while, do, switch**).

Tutte le istruzioni del blocco devono rientrare: 2-3 colonne dovrebbe essere una misura corretta, purché venga usata sempre la stessa quantità.

Un'indentazione esagerata, lascia poco spazio agli indispensabili commenti.

```

#include <stdio.h>
#define FIGRECO 3.14159
int Sub1 ( float );
/*=====
...specifiche del Software, autori, data release...
-----*/

int main ( void )
{ float Area,      // ... uso della variabile
  Raggio;         // ...
  int   i,         // ...
        j;        // ...

  istruzione;
  do
  { istruzione;
    if ( condizione ) // motivo della selezione
    { istruzione;
      istruzione;
      for ( i = 0, j = 100; i < j ; i ++, j -= 13)
      { istruzione;
        istruzione;
      }
    }
    else // alternativa
    { istruzione;
      istruzione;
      Area = Raggio > 0 ? Raggio * Raggio * FIGRECO : 0;
    }
  }
  while (condizione);
}

```

Main.c

```

#include <stdio.h>

/*=====
...specifiche della Subroutine, data ultima modifica...
INPUT:  parametri di ingresso
OUTPUT: valore di uscita
-----*/

int Sub1
( float x      // .....
)
{ int k;       // .....

  istruzione;

  switch ( k )
  { case 1 : istruzione;
    istruzione;
    break;

    case 23: istruzione;
    istruzione;
    break;

    case 4 : istruzione;
    istruzione;
    break;

    default: istruzione;
  }

  return valore;
}

```

Sub1.c