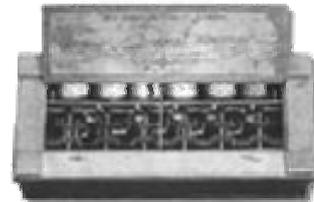


TURBO PASCAL ver. 7.0

Il linguaggio **PASCAL** prende il nome da **BLAISE PASCAL** (matematico e filosofo del XVII sec.) inventore della **PASCALINA** (la prima macchina calcolatrice).



Blaise Pascal (1623-1662)



1642- La pascalina

Il linguaggio venne elaborato verso la fine degli anni '60 da **NIKLAUS WIRTH**, docente all'università di Zurigo.



Niklaus Wirth

Nel 1983 una società di software, la Borland International, commercializzò un compilatore del Pascal chiamato Turbo Pascal.

Il PASCAL è:

- ✓ un linguaggio **evoluto**.
- ✓ un linguaggio di programmazione **strutturato** perché si basa su strutture logiche e organizzativi precise.
- ✓ un **insieme di caratteri ASCII**: lettere, numeri, simboli, ritorni a capo e spazi.

Tabella dei caratteri ASCII

0		43	+	86	V	129	ü	172	¼	215	‡
1	☺	44	,	87	W	130	é	173	ı	216	‡
2	●	45	-	88	X	131	â	174	«	217	ı
3	♥	46	.	89	Y	132	ä	175	»	218	ı
4	♦	47	/	90	Z	133	à	176	■	219	■
5	♣	48	0	91	[134	á	177	■	220	■
6	♠	49	1	92	\	135	ç	178	■	221	■
7	•	50	2	93]	136	ê	179		222	■
8	◼	51	3	94	^	137	ë	180	ı	223	■
9	○	52	4	95	_	138	è	181	ı	224	α
10	◻	53	5	96	'	139	ï	182	ı	225	β
11	♂	54	6	97	a	140	î	183	ı	226	Γ
12	♀	55	7	98	b	141	ì	184	ı	227	π
13	♪	56	8	99	c	142	Ë	185	ı	228	Σ
14	♫	57	9	100	d	143	Å	186	ı	229	σ
15	☒	58	:	101	e	144	É	187	ı	230	μ
16	▶	59	;	102	f	145	æ	188	ı	231	τ
17	◀	60	<	103	g	146	Æ	189	ı	232	Φ
18	↕	61	=	104	h	147	ô	190	ı	233	Θ
19	!!	62	>	105	i	148	ö	191	ı	234	Ω
20	¶	63	?	106	j	149	ò	192	ı	235	δ
21	§	64	@	107	k	150	û	193	ı	236	∞
22	■	65	A	108	l	151	ù	194	ı	237	∅
23	↕	66	B	109	m	152	ÿ	195	ı	238	ε
24	↑	67	C	110	n	153	ÿ	196	-	239	∩
25	↓	68	D	111	o	154	Ü	197	ı	240	≡
26	→	69	E	112	p	155	ç	198	ı	241	±
27	←	70	F	113	q	156	ε	199	ı	242	≥
28	└	71	G	114	r	157	¥	200		243	≤
29	┌	72	H	115	s	158	₤	201	ı	244	
30	▲	73	I	116	t	159	ƒ	202	ı	245	ı
31	▼	74	J	117	u	160	á	203	ı	246	÷
32		75	K	118	v	161	í	204	ı	247	≈
33	!	76	L	119	w	162	ó	205	=	248	°
34	"	77	M	120	x	163	ú	206	ı	249	•
35	#	78	N	121	y	164	ñ	207	ı	250	·
36	\$	79	O	122	z	165	Ñ	208	ı	251	√
37	%	80	P	123	{	166	ª	209	ı	252	n
38	&	81	Q	124		167	º	210	ı	253	²
39	'	82	R	125	}	168	ç	211	ı	254	▪
40	(83	S	126	~	169	ı	212	ı	255	
41)	84	T	127	Δ	170	ı	213	ı		
42	*	85	U	128	ç	171	½	214	ı		

Per ottenere un carattere della tabella basta premere il tasto Alt e digitare il codice corrispondente sul tastierino numerico nella parte destra della tastiera.

Dagli anni '60 ad oggi si sono avute parecchie revisioni di questo linguaggio, l'ultima è la versione **7.0** di **TURBO PASCAL**.

Il Turbo Pascal è un **ambiente di lavoro integrato**, cioè è possibile scrivere, memorizzare, compilare ed eseguire un programma senza uscire dall'ambiente di programmazione; i collegamenti tra l'ambiente EDIT, quello di compilazione e quello di esecuzione vengono fatti automaticamente dal computer.

Nel **TURBO PASCAL** vi sono quattro classi di elementi:

- ✓ Le parole riservate.
- ✓ Gli identificatori.
- ✓ Le costanti.
- ✓ I simboli.

Le parole riservate, dette *standard*, sono:

AND	ARRAY	BEGIN	CASE
CONST	DIV	DO	DOWNTO
ELSE	END	FILE	FOR
FORWARD	FUNCTION	GOTO	IF
IN	LABEL	MOD	NIL
NOT	OF	OR	PACKED
PROCEDURE	PROGRAM	RECORD	REPEAT
SET	THEN	TO	TYPE
UNTIL	VAR	WHILE	WITH

Nel **Turbo Pascal** sono presenti anche le seguenti parole riservate:

ABSOLUTE	ASM	DESTRUCTOR
IMPLEMENTATION	INTERFACE	OBJECT
PRIVATE	SHR	UNIT
VIRTUAL	CONSTRUCTOR	EXTERNAL
INLINE	INTERRUPT	SHL
STRING	USES	XOR

Gli identificatori

Gli **identificatori** sono parole (diverse da quelle riservate) che servono per indicare grandezze variabili e costanti, tipi di dati, procedure, funzioni e alcuni altri dati.

Esistono vari tipi di identificatori:

- predefiniti dal Turbo Pascal
- definiti dal programmatore.

Alcuni degli identificatori predefiniti sono: integer, real, byte, sin, ...

Gli identificatori definiti dal programmatore sono gli elementi aggiunti al linguaggio come variabili, procedure, funzioni, etc..

Un **identificatore** è una sequenza da 1 a 127 caratteri, che inizia con una lettera, non contiene spazi vuoti o simboli speciali: &, !, *, etc., può contenere il trattino_.

Per il Turbo Pascal non esistono differenze tra maiuscole e minuscole. Tutti gli identificatori in Pascal devono essere definiti prima di utilizzarli. Sono parole identificatrici, ad esempio, **Area** oppure **Area_triangolo**.

Gli operatori

ELENCO DEGLI OPERATORI ALGEBRICI

+	somma
—	sottrazione
/	divisione
*	moltiplicazione
div	divisione fra interi
mod	modulo

ELENCO DEGLI OPERATORI RELAZIONALI

<	essere minore di
>	essere maggiore di
=	essere uguale a
> =	essere maggiore o uguale a
< =	essere minore o uguale a
< >	essere diverso da

ELENCO DEGLI OPERATORI LOGICI

not	negazione
and	prodotto logico
or	somma logica

Tipi di dati

La gestione dell'informazione in Turbo Pascal si realizza mediante diverse classi di dati.

I principali tipi semplici sono:

Integer	Numeri interi senza parte decimale.
Char	Caratteri del codice ASCII
Boolean	Possono contenere i valori di falso o vero
Real	Numeri che possono avere una parte decimale
String	Una sequenza di caratteri che viene trattata come un solo dato.

Variabili e costanti

I tipi di dati che utilizziamo nei nostri programmi possono essere di due classi: **variabili** o **costanti**.

Come indicato dal nome le variabili possono cambiare contenuto nel corso dell'esecuzione del programma, mentre le costanti mantengono il loro valore fisso durante tutto il processo (un'esecuzione del programma). In sostanza a differenza delle variabili, il nome di una costante si identifica col suo valore.

Le variabili e le costanti possono essere di tutti i tipi visti precedentemente: numeriche tanto intere che reali, caratteri, stringhe di caratteri, etc.

Commenti

E' possibile introdurre commenti in un nostro programma utili a migliorare la comprensione del codice sorgente, quindi è auspicabile.

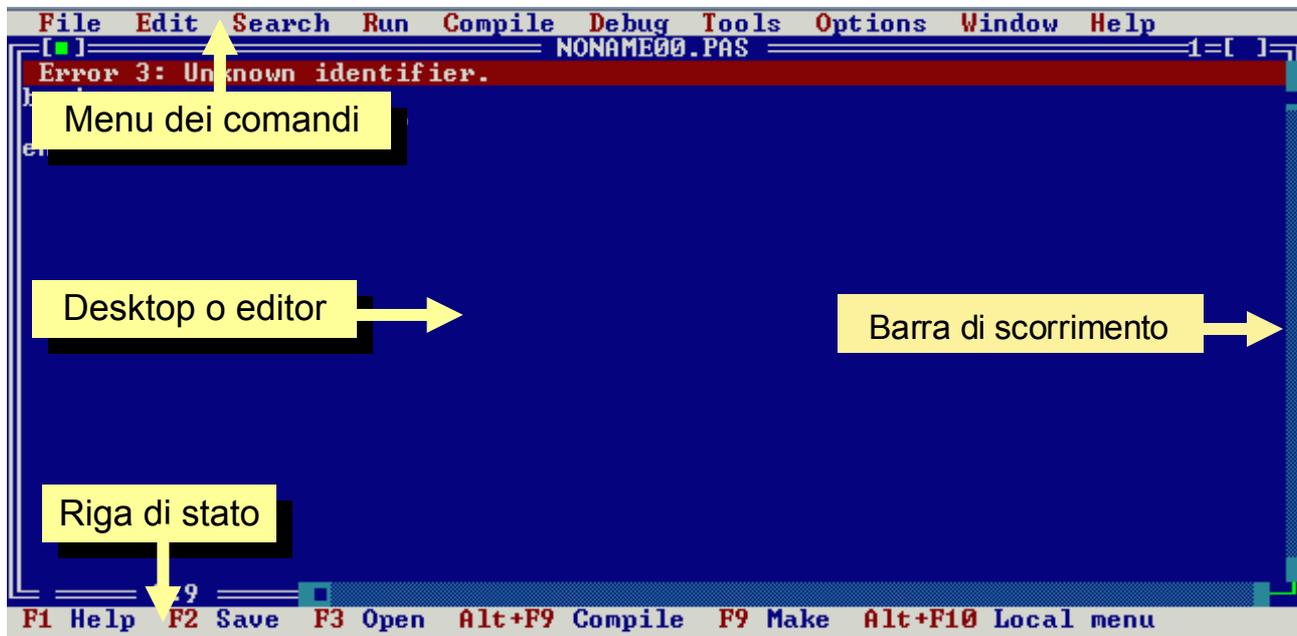
I commenti sono importanti al momento della compilazione del programma e sono di enorme importanza al momento della creazione, della modifica o della manutenzione.

Ci sono due forme di inserimento dei commenti in un programma Turbo Pascal, usare le graffe: {Commento} o le parentesi con asterisco: (*Commento*).

Ambiente di lavoro Turbo Pascal

Una volta avviato il TurboPascal si presenterà un'interfaccia a menu abbastanza intuitiva.

Compariranno vari menu, ma quelli principali sono tre: **File**, **Run**, **Compile**.



Il menu **File** è formato da **New**, per creare un nuovo documento di lavoro; **Load**, per aprire lavori già salvati; **Save**, registra un lavoro.

File	
Open...	F3
New	
Save	F2
Save as...	
Save all	
Change dir...	
Print	
Get info...	
DOS shell	
Exit	Alt-X

Nel menu **Compile** sono presenti il comando **Compile**, per compilare il programma e il comando **Destination**, per decidere se compilare il programma nella Ram oppure in un file eseguibile(es. nome_programma.exe).

Compile

Compile	Alt-F9
Make	F9
Duild	
Destination	Memory
Primary file...	

Una volta compilato il programma, questo deve essere eseguito, e perciò è necessario selezionare dal menu **Run** il comando **Run**.

Run

Run	Ctrl-F9
Program reset	Ctrl-F2
Go to cursor	F4
Trace into	F7
Step over	F8
Parameters...	

La scrittura di un programma

❖ 1° passo

Scrivere l'algoritmo risolutivo in linguaggio di progetto.

❖ 2° passo

Tradurre l'algoritmo in linguaggio pascal (**CODIFICA**) mediante l'editor di testi del pascal → si ottiene uno o più file che formano il **CODICE SORGENTE**.

Esempio

Vogliamo che il PC risolva il seguente problema:

"Dato un numero naturale trovare il suo successivo"

L'algoritmo, scritto in linguaggio di progetto, che risolve il problema è:

```

algoritmo NumeroSuccessivo;
var Numero, Successivo:
intero;
inizio
  scrivi introduci il numero;
  leggi Numero;
  Successivo:= Numero+1;

```

L'algoritmo tradotto in linguaggio Pascal è:

```

program NumeroSuccessivo;
uses crt;
var Numero, Successivo: integer;
begin
  write ('introduci il numero:');
  readln (Numero);
  Successivo:= Numero+1;
  writeln (Successivo);
  readln

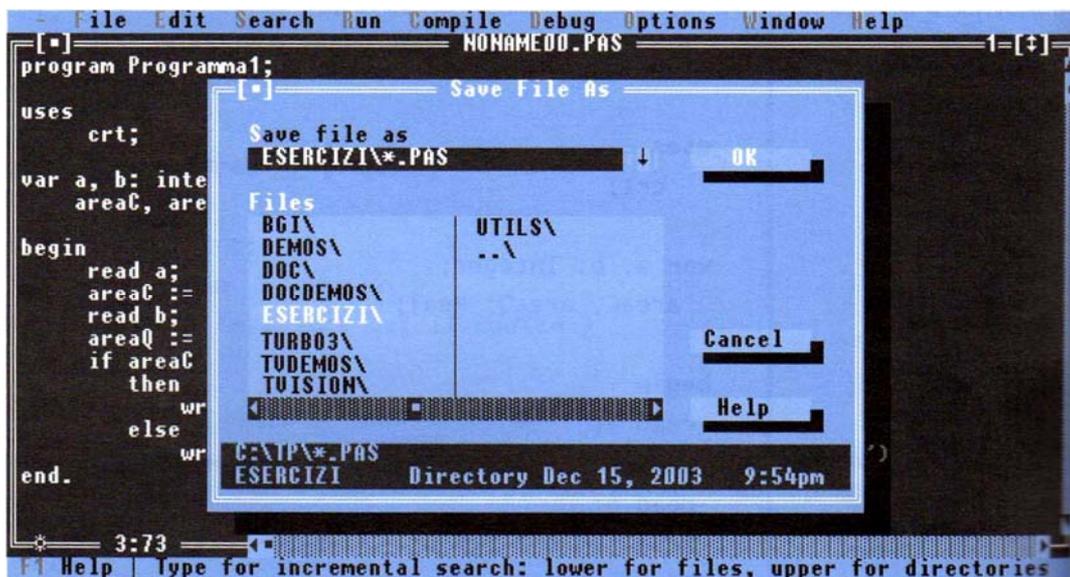
```

Dopo aver aperto il Turbo Pascal, scrivere nell'editor il programma.

FILE → **NEWS** → si apre la finestra di Edit di nome **NONAME00.PAS** (è il nome che per default sarà assegnato al file del programma che verrà digitato nella finestra).

Memorizzazione del programma

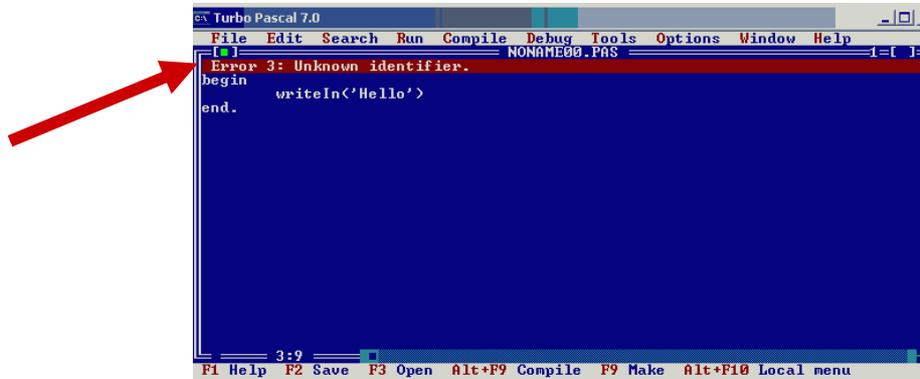
FILE → **CHANGE DIR** → scegliere la directory o il disco e inviare →
FILE → **SAVE** (la prima volta chiede di assegnare il nome al file) oppure premere il tasto **F2**.



Compilazione del programma

COMPILE → **COMPILE** [ALT+F9].

Se il compilatore trova errori di sintassi, la compilazione viene interrotta e l'errore segnalato con un messaggio sul video.



Il programma compilato prende il nome di **codice oggetto** o **codice compilato**.

Esecuzione di un programma

RUN → **RUN** [CTRL+F9]

Se il PC è molto veloce per vedere l'output del programma premere **ALT+ F5** oppure **DEBUG** → **USER SCREEN**

Tipi di errore

Nel digitare un programma si possono commettere diversi generi di errore.

- ❖ **Errori di sintassi:** sono rilevati dal compilatore (ad es. manca un punto e virgola, non è chiusa una parentesi, ...)
- ❖ **Errori di run-time:** emergono solo in fase di esecuzione (ad es. divisione per zero, radice quadrata di un numero negativo, ...)
- ❖ **Errori di logica** (portano a risultati diversi da quelli voluti e non sono rilevati dal compilatore)

Esempio di errore di sintassi

```
program errore1;
uses crt;
var numero, cubo:integer;
begin
  readln(numero);
  cubo:=numero*numero*numero
  writeln(cubo);
  readln;
end.
```

Esempio di errore RUN TIME

```
program errore2;
uses crt;
var x, reciproco: real;
begin
  readln(x);
  reciproco :=1/x;
  writeln(reciproco);
  readln;
end.
```

Esempio di errore logico

```
program errore3;
uses crt;
var voto1, voto2, voto3, media: real;
begin
  readln(voto1, voto2, voto3);
  media := voto1+ voto2+ voto3/3;
  writeln(media);
  readln;
end.
```

Correzione errori di sintassi

➤ Per cancellare un carattere

Posizionare il cursore sul carattere e premere il tasto **CANC**

Oppure posizionare il cursore subito dopo il carattere e premere il tasto **BACKSPACE**

➤ Per cancellare una riga

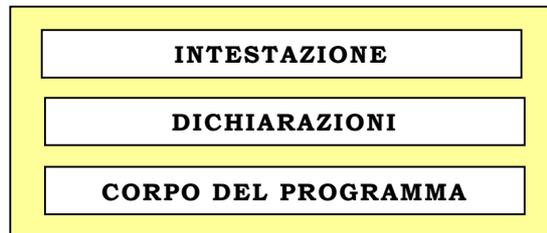
Posizionare il cursore sulla riga e premere i tasti **CTRL+Y**

➤ Per inserire una riga vuota

Posizionare il cursore dopo l'ultimo carattere della riga precedente e premere **INVIO** oppure premere i tasti **CTRL+N**

La struttura di un programma in Pascal

Un programma, scritto in linguaggio Turbo PASCAL, si compone delle seguenti parti:



Intestazione (indica il nome del programma)

Program <nome programma>;

E' una riga che inizia con la **parola riservata** (keyword) **program** seguita dal nome che si vuol dare al programma.

Dopo il nome del programma deve essere messo un ; (punto e virgola). Il punto e virgola è molto usato in Pascal (ed in genere da molti linguaggi di programmazione): serve a separare due istruzioni.

Esempi:

```
program prova;  
program primoProgramma;  
program primo_programma;  
program capitolo1;
```



Sezione dichiarativa (indica gli oggetti e il tipo di oggetti usati)

Qui il programmatore assegna un nome (identificatore) ai dati che devono essere memorizzati durante l'elaborazione.

La parte dichiarativa comprende:

a) dichiarazione di costante

const <identificatore>=<valore>;

La dichiarazione delle costanti viene fatta prima della dichiarazione di variabile mediante la parola riservata **const** seguita dal nome della costante e dal suo valore. Il valore può essere sia un numero sia una successione di caratteri racchiusa tra apici.

Esempi:

```
const Area=10;  
const nome='Maria';
```

Il Pascal permette l'uso di due costanti predefinite, **PI**=3.1415926536 e **Maxint**=32767, che non necessitano di dichiarazione ma possono essere usate direttamente in un programma mediante il loro identificatore.

Esempio:

```
Circonferenza:=PI*Diametro;
```

b) dichiarazione di variabile

```
var <identificatore>:<tipo>;
```

Tutte le variabili usate nel programma devono essere definite. Dopo aver attribuito un nome (o identificatore) alla variabile, è necessario associare ad essa un **tipo**, cioè un insieme di valori che la variabile può assumere e l'insieme di tutti gli operatori ammessi in tale insieme.

Esempi:

```
var Area:real;  
var A, B:integer;
```

Tipi di variabili:

- numeriche Esempio: **var** a, b:**integer**;
- caratteri (lettere, cifre, caratteri speciali racchiusi tra apici)
Esempio: **var** iniziale:**char**;
- stringhe (successione di caratteri di lunghezza definita tra []; il loro valore deve essere racchiuso tra apici)
Esempio: **var** nome:**string [10]**;
- booleane (possono avere solo due valori, vero/falso ovvero true/false)
Esempio: **var** segnale:**boolean**;

c) dichiarazione di funzione e di procedura

```
function < nome della funzione>(<parametri>:<tipi dei parametri>):<tipo della  
funzione>;  
begin  
  <corpo della funzione>  
end.
```

```
Procedure <nome della procedura> (<parametri>:<tipi dei parametri>);  
begin  
  <corpo della procedura>  
end.
```

Le procedure e le funzioni sono programmi che rendono più agevole il lavoro di programmazione. Esistono procedure standard incorporate nel Pascal. Le procedure e le funzioni possono anche essere create dal programmatore.



Sezione esecutiva (corpo principale del programma)

È la parte operativa del programma. Corrisponde in pratica alla traduzione dell'algoritmo. La parte dichiarativa è infatti solo preparatoria e serve al compilatore per essere più efficiente (ad esempio se conosce in anticipo i tipi delle variabili può organizzarle più efficientemente nella RAM) e controllare il resto del programma per alcuni tipi di errore (ad esempio tentare di modificare una costante). Questa sezione inizia con la parola riservata **begin** e termina con la parola riservata **end** seguita da un punto.

Per rendere semplice la lettura dei programmi viene adottata la seguente convenzione:

- le parole riservate verranno scritte in **neretto minuscolo**;
- gli identificatori per costanti e variabili verranno scritti in carattere **stampatello maiuscolo**;
- si userà il metodo della **indentazione** nella descrizione di ogni programma.

Esempio 1

➤ In **linguaggio di progetto** l'algoritmo è:

```
inizio
  leggi (N);
  N:=N*2;
  scrivi (N);
fine.
```

➤ In **linguaggio pascal** è:

```
program DOPPIO; _____ sezione intestazione
var _____ sezione dichiarazioni
  N: integer; _____
begin
  readln(N); _____
  N:=N*2; _____
  writeln(N); _____
end. _____
```

corpo del programma

Il simbolo ";" che si trova alla fine di ogni riga, indica la fine di una istruzione.
Il simbolo "." indica invece la fine del programma.

Esempio 2

program RETTANGOLO;	_____	intestazione
uses crt;	_____	dichiarazione della libreria crt
(*determina l'area del rettangolo*)	_____	riga di commento
var BASE,ALTEZZA,AREA: integer ;	_____	dichiarazione delle variabili
begin	_____	inizio del corpo del programma
clrscr ;	_____	istruzione per pulire lo schermo
writeln ('introduci misura base');	_____	istruzione di scrittura
readln (BASE);	_____	istruzione di lettura
writeln ('introduci misura altezza');		
readln (ALTEZZA);		
AREA := BASE *ALTEZZA;	_____	istruzione di assegnazione
writeln ('l'area è ',AREA);		
readln		
end.	_____	fine del corpo del programma