

Informatica (corso di Laurea in Ing. Chimica) A.A. 2001/02

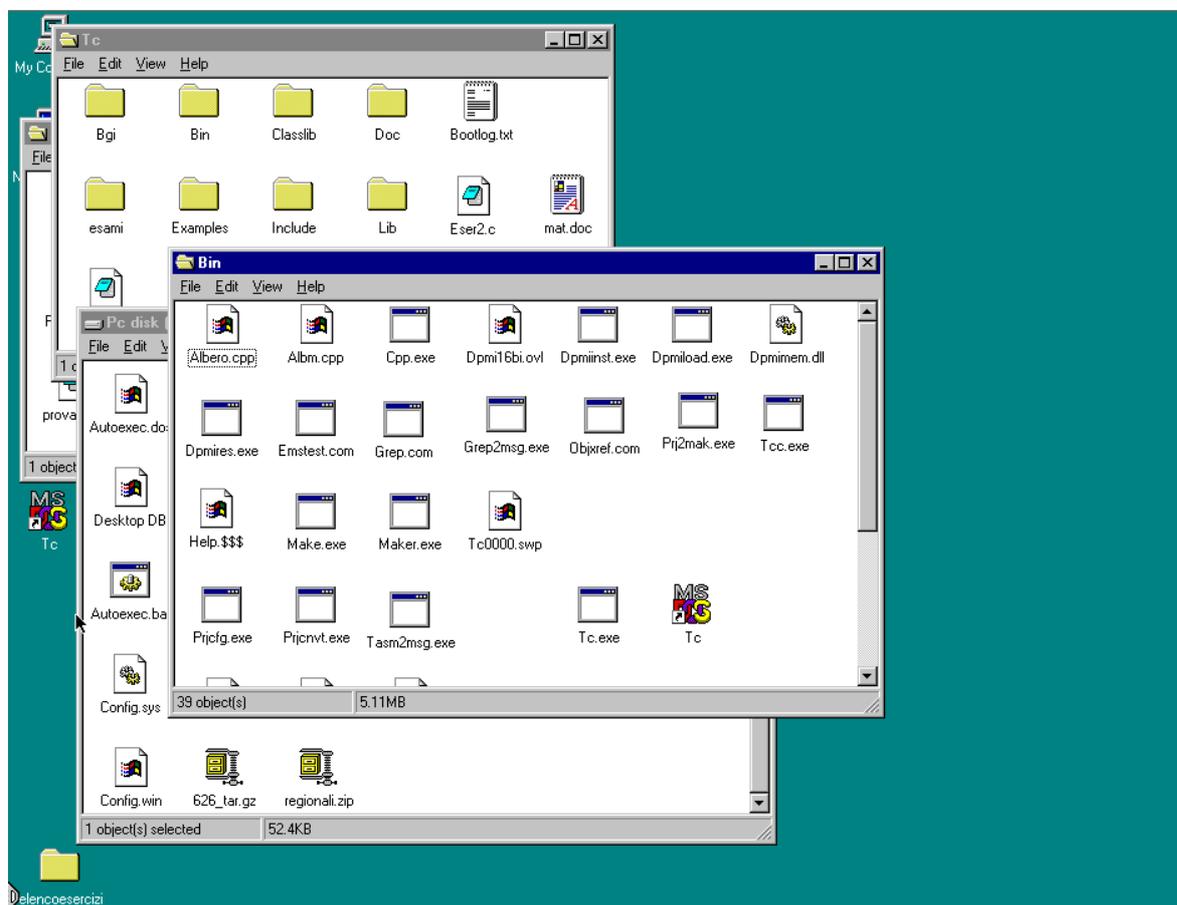
Prima esercitazione autoguidata

1.1. Aspetto di Windows

Acceso il calcolatore, dopo il *bootstrap* (che corrisponde alla fase di predisposizione al funzionamento), è attivo un particolare programma: il Sistema Operativo.

Un sistema operativo è un insieme di programmi eseguibili che funge da interfaccia tra l'utente e il calcolatore. Esso intercetta tutte le richieste fatte dall'utente, relativamente all'uso di risorse disponibili nel calcolatore, e le soddisfa (se possibile ...).

In laboratorio sono disponibili macchine dotate di sistema operativo WINDOWS in due versioni (NT e 2000). Nel seguito faremo riferimento a WINDOWS NT, ma tutto quel che diciamo può



essere utilizzato per 2000, magari con qualche ovvia variazione.

L'utente comunica le proprie richieste utilizzando due principali strumenti di *input*: la tastiera e il mouse. In un sistema operativo come WINDOWS, l'uso del mouse è importante: le risorse del calcolatore sono rappresentate per lo più da figure che appaiono sullo schermo (le *icone*) e l'uso di queste risorse viene richiesto dall'utente mediante il mouse. Con la tastiera le "richieste di uso di risorse" vengono impartite al calcolatore per iscritto: è cioè possibile comunicare le proprie richieste mediante un apposito linguaggio di comandi.

La figura rappresenta un esempio di immagine video durante una sessione di lavoro in WINDOWS. Oltre alle icone, notiamo anche dei contenitori di icone, le *finestre*. In figura sono rappresentate due

finestre. Una finestra è uno spazio delimitato all'interno dello schermo. Ad ogni finestra è assegnato un nome.

Nelle varie finestre è possibile individuare i due tipi di icone fondamentali, l'icona corrispondente ad un FILE di dati e quella corrispondente ad una DIRECTORY.

I file (e quindi le directory) contengono tutte le informazioni disponibili al calcolatore nelle sue unità di memoria secondaria (hard disk, floppy disk, CD ...).

L'organizzazione di queste informazioni e la loro reperibilità è amministrata dal File System che è una delle componenti del sistema operativo.

1.2. File System

File

Un blocco unitario di locazioni in memoria secondaria, contenente informazioni omogenee e contraddistinto da un nome.

Directory

Contenitore che raggruppa sotto uno stesso nome più file (e altre directory)

Il nome del file

Il nome di un file può essere semplicemente una sequenza di caratteri. Di solito però è del tipo
Nomefile.estensione

Ad esempio,

Tc.exe, Bootlog.txt, Autoexec.bat, eser2.c, mat.doc

L'estensione caratterizza il tipo di dati contenuti nel file.

Nel vecchio sistema operativo DOS (base per lo sviluppo degli odierni sistemi operativi Windows), nomefile doveva essere di al più 8 caratteri e l'estensione di 3. In WINDOWS l'estensione può anche non esserci. Se c'è, nella maggioranza dei casi viene usata per associare una icona particolare al file. Ad esempio

Tc.exe ha un'icona che permette di individuarlo immediatamente come file eseguibile.

mat.doc ha un'icona che lo evidenzia come file "word" (cioè da usare con un particolare programma di word processing che si chiama WORD).

Una volta associata un'icona ad un tipo di file, diventa superfluo mostrare l'estensione che potrebbe quindi essere omessa nella visualizzato: sullo schermo potremmo vedere ad esempio i nomi mat o Tc. Per i file cui non è associabile un'icona prestabilita se ne usa uno standard.

Le unità di memoria secondaria

Ogni unità di memoria secondaria ha un **nome logico** che corrisponde ad un "indirizzo" associato con quella unità e che permette di individuarla in modo univoco.

- A: è il nome logico dell'unità di gestione dei floppy disk
- B: è il nome logico di una eventuale seconda unità di gestione dei floppy disk
- C: è il nome logico dell'hard disk Pc disk (C:) nella figura sotto
- D: E: ... tutte le altre lettere vengono usate per le eventuali altre periferiche di memoria connesse al calcolatore (altri hard disk, unità di gestione per i CD, zip, etc).

L'unità C : è particolarmente importante, perché di solito si riferisce all'unità disco che contiene anche il sistema operativo. Useremo C: per spiegare il path di identificazione di un file. Tutto quel che diremo vale anche per le altre "lettere" ...

Identificazione di un file

Dato un file, questo sarà contenuto in una certa unità di memoria secondaria.

Spesso per usare un file non basta "cliccare" col mouse, ma bisogna scrivere esplicitamente il suo nome. E quasi sempre scrivere solo *nomefile* e eventualmente *estensione* non basta. Infatti va specificato con precisione "dove" il file si trova all' interno dell'unità di memoria secondaria in cui è contenuto. Ciò per "individuare" univocamente, distinguendolo da eventuali altri file che abbiano stesso nome e estensione.

Evidentemente la prima informazione utile all' individuazione del file è il nome dell'unità di memoria in cui esso è contenuto.

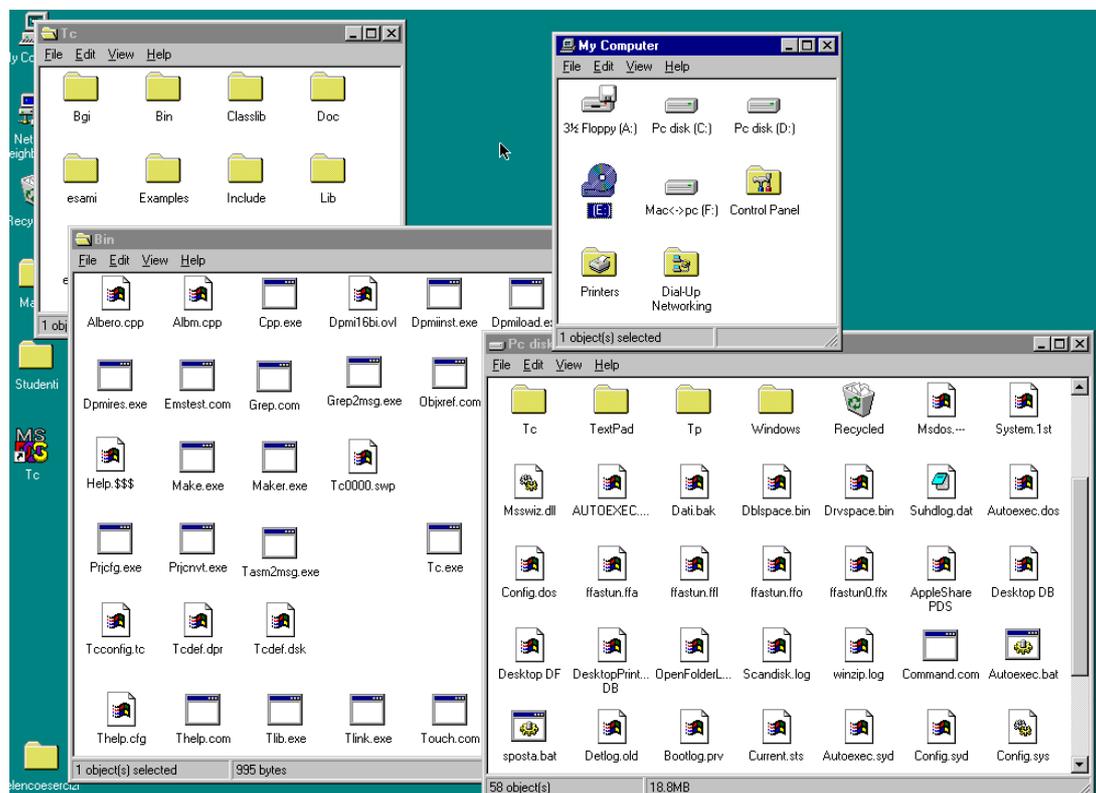
Ad esempio l' individuazione esatta del file `autoexec.bat` della prossima figura è data dal suo nome e dal fatto che sta nel disco C:. Il suo nome completo è pertanto `C:\autoexec.bat`.

Sappiamo però che un file potrebbe essere contenuto in una directory, e questa potrebbe essere contenuta in un' altra e così via.

Ad esempio, il file `tc.exe` (la ' t ' maiuscola o minuscola non ha importanza) risiede nella directory BIN, che risiede nella directory TC che sta in C :.

Il nome completo del file in questione è `C:\TC\BIN\tc.exe`. Il *percorso* di directory e nome logico dell'unità che permette di individuare il file si chiamano **path**. Nel path i singoli nomi sono separati dal carattere '\' (backslash).

Nella figura seguente è mostrata la posizione del file, e la directory TC appare dentro la finestra Pc disk (C:). (curiosità: Windows è proprio la directory in cui sono contenuti tutti i file/programmi che compongono il sistema operativo.)



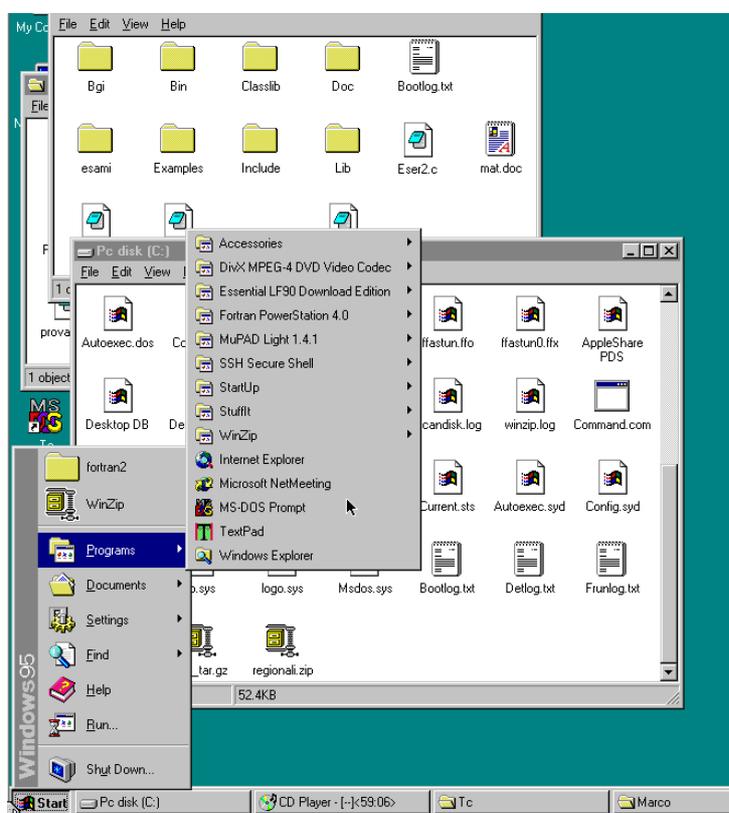
Per ricapitolare: se vogliamo identificare un file, bisogna che sia chiaramente indicato non solo il nome (NOMEFILE seguito dalla possibile ESTENSIONE), ma anche tutto il percorso di directory che ci permettono di arrivare al file, a partire da C: (o dall'unità di memoria secondaria in cui il file è memorizzato).

1.3. DOS

Il Dos è il sistema operativo adottato dalle macchine tipo PC fino a non molti anni fa. È comunque presente anche nei calcolatori su cui è installato WINDOWS, il quale deve interagire costantemente con DOS. Ad esempio il modo di indirizzare i file con percorso\NOMEFILE.EXT discende direttamente dal DOS. Inoltre quando useremo il TurboC dovremo essere in grado di fare cose elementari con il DOS (come ad esempio scrivere il nome del file in cui vogliamo scrivere il programma), e potremo occasionalmente "entrare nella DOS-shell" per compiere delle operazioni. Infatti una volta entrati nel programma TurboC (TC) non è possibile interagire con Windows se non uscendo momentaneamente dal TC, mentre invece è possibile interagire con il Dos entrando in una finestra (chiamata appunto *Dos shell*) per poi tornare nell'ambiente TC dove tutto è rimasto immutato. Perché uno/una debba interagire con il sistema operativo mentre scrive programmi ... sarà chiaro in seguito.

Finestra DOS (DOS Prompt)

I movimenti del mouse si comunicano ad un cursore (a forma di freccia, di solito) che appare sullo schermo. Muovete il mouse e individuate il cursore, se non lo avete già fatto). Quando la freccia è posizionata su una icona, premendo il pulsante sinistro del mouse, l'icona cambia colore: si dice che è *selezionata*.



Selezionare l'icona a forma di pulsante in basso a sinistra dello schermo, su cui è scritto AVVIO (START). Quel che succede è definito "apertura di un menù a tendina". Viene infatti visualizzata una colonna di nomi (ciascuno anche accoppiato ad una piccola icona). Muovendo il mouse in alto e in basso (senza bisogno di cliccare) si selezionano le varie voci. Selezionare la voce PROGRAMMI (PROGRAMS). Si apre un altro menù, sul quale bisogna trasferire la freccia muovendo il mouse per selezionare la voce PROMPT DI MS-DOS. A questo punto si apre una finestra particolare sul video, all'interno della quale è possibile far eseguire comandi del sistema operativo DOS

La sequenza di caratteri `C:\WINDOWS>` si chiama prompt in quanto indica la disponibilità del sistema operativo a ricevere comandi.

Incidentalmente fornisce anche informazioni sulla situazione relativa al file system: su quale unità di memoria secondaria siamo "posizionati" e, in essa, all'interno di quali directory.

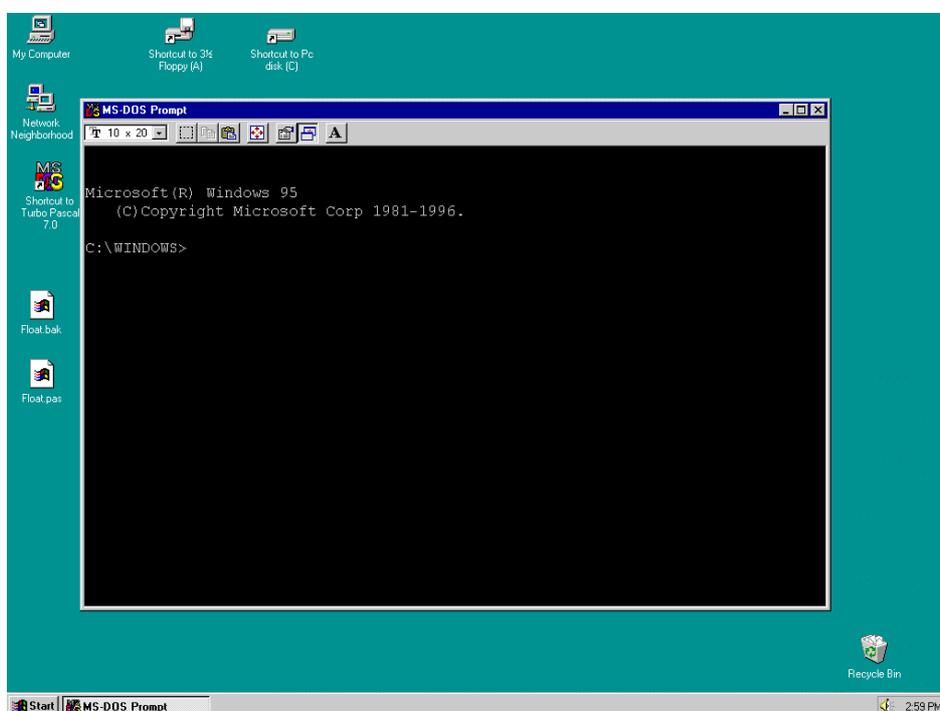
Nella figura precedente, DOS ci dice che possiamo inserire comandi e che siamo posizionati all'interno della directory WINDOWS dentro al disco C:.

I comandi, in un sistema operativo come DOS, vanno scritti usando la tastiera e "inviati", cioè comunicati al DOS perché li esegua. Questo invio avviene premendo il tasto di return, dopo che la scrittura del comando è stata completata. Il nome o il simbolo presente sul tasto di return potrebbe esser Ret, "↵" o Return, o Enter ...

Alcuni comandi di base del DOS:

DIR: visualizza il contenuto della directory corrente. Ad esempio, nel caso della figura precedente, se scriviamo DIR e premiamo il tasto di return, il calcolatore stampa sul video la lista di tutti i file e directory presenti nella directory WINDOWS in C:.

DIR "PATH" Dir è un comando elastico e accetta anche l'aggiunta di una specifica di path. In tal caso visualizza sul video il contenuto della directory individuata dal path. Ad esempio DIR A:\fonda\lez1 ci permette di vedere il contenuto della directory lez1 contenuta nella directory fonda nell'unità floppy (dischetto) A:.



CD: abbiamo detto che il prompt indica la directory sulla quale siamo posizionati. CD permette di cambiare tale posizionamento. Ad esempio l'esecuzione di CD C:\TC modifica la directory corrente, in modo che sia la TC dentro a C:. Come conseguenza il prompt sarà modificato in C:\TC>

Non sempre è necessario scrivere un path completo (anzi quasi mai ...). Se siamo posizionati in C:\TC e eseguiamo il comando cd BIN, la directory BIN diventa quella corrente. Il prompt si trasforma in C:\TC\BIN.

CD, Cd, cd sono la stessa cosa.

CD .. (CD seguito da due punti) "sposta all'indietro" (si dice anche che "risale") di un passo nella gerarchia delle directory. La directory che contiene quella in cui siamo posizionati diventa corrente. Se siamo posizionati in C:\TC\BIN ed eseguiamo CD .. il prompt diventa C:\TC> (e TC è la nuova directory corrente). Se eseguiamo CD .. ancora una volta arriviamo alla "radice" dell'unità C: (il prompt diventa C:\>). È impossibile risalire più indietro.

Per cambiare l'unità disco su cui siamo posizionati, è sufficiente scrivere la lettera che indica l'unità desiderata seguita dai due punti. Per esempio il comando **A:** ci posiziona sull'unità A: cioè sul gestore (*driver*) dei floppy disk (il prompt cambia in A:\>). Se poi facciamo DIR, vediamo il contenuto del floppy disk (se è inserito).

FORMAT è il comando per formattare le unità di memoria secondaria (soprattutto i floppy disk). L'operazione di formattazione "predispone l'uso del disco come supporto di memoria secondaria": si controlla l'integrità fisica del supporto; si predispongono i dati relativi all'organizzazione del disco (ad esempio viene "creata" la directory radice, come A).

FORMAT A: oppure FORMAT A:\ formatta il disco contenuto nell'unità A: viene richiesto di

inserire un dischetto nel drive A:

e premere un tasto quando fatto poi inizia la formattazione

... formattazione in corso ...

poi viene richiesto di dare un nome al dischetto (label = etichetta)

label (11 caratteri che etichettano il disco)

poi vengono date alcune informazioni e si passa, se necessario a formattare un successivo dischetto

altro disco ?

(se dobbiamo formattare altri dischi rispondiamo positivamente, altrimenti negativamente).

Come risultato, il contenuto attuale del dischetto viene cancellato, le tracce ed i settori del disco vengono predisposti per contenere dati secondo le modalità previste per un PC.

COPY <NOMEFILE1> <NOMEFILE2>

copia il file <NOMEFILE1> nel file <NOMEFILE2>

DEL <NOMEFILE>

cancella il file <NOMEFILE>. (non è più possibile recuperarlo, quindi attenti alle cancellazioni).

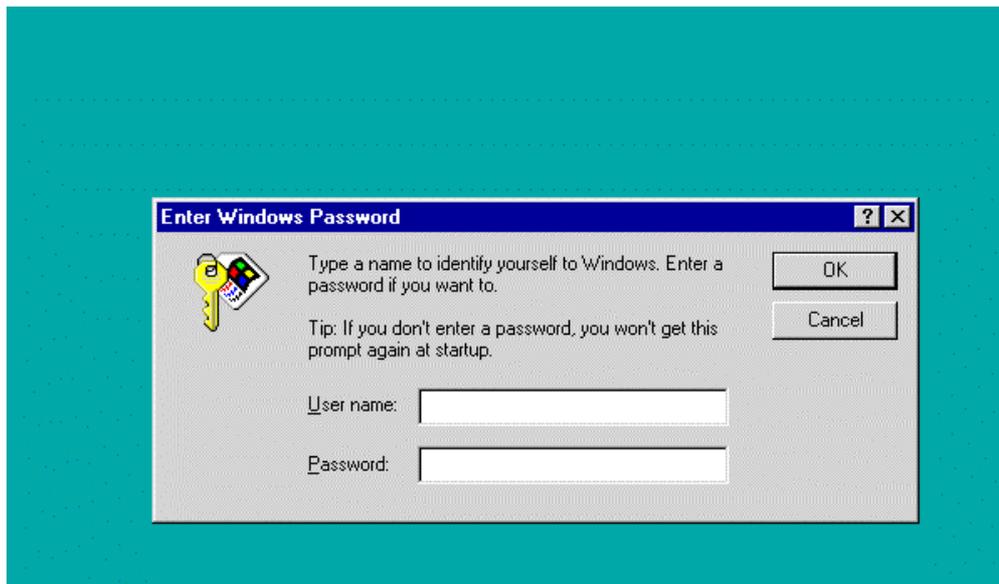
Per tutti gli altri comandi significativi e per i dovuti approfondimenti, conviene guardare la dispensa di DOS consigliata nel materiale del corso.

1.4. Accensione del PC e accesso al TURBOC

I Personal Computer sui quali vengono svolte le esercitazioni hanno due interruttori separati: uno sul video ed uno nella parte posteriore dell'unità base. Una volta acceso, il computer inizia una serie di operazioni di autocontrollo della memoria installata, di caricamento del sistema operativo WINDOWS NT e di alcuni programmi necessari per il collegamento in rete.

Alla fine di questa fase appare a tutto schermo la "scrivania di lavoro", contenente riferimenti grafici (Icône) a vari software disponibili e a directory (Folder) di memoria secondaria.

Questo potrebbe essere l'aspetto del calcolatore all'inizio della sessione di lavoro (calcolatore già acceso). C'è una finestra di login centrata sullo schermo.



L'accesso al calcolatore è permesso solo agli utenti autorizzati (dotati di un *account*).

Normalmente, l'utente dotato di un account scrive la propria *username* (nome caratteristico dell'utente) e scrive la propria *password* (cioè la parola che, in teoria è nota solo all'utente e senza la quale non sarà consentito l'accesso).

Notare che potrebbe esserci un terzo spazio bianco per il *workgroup* (nome del gruppo di lavoro, non in figura). Questo dipende dalla configurazione delle macchine a disposizione. Nel caso della macchina da cui è stata presa l'immagine qui mostrata il workgroup non era previsto.

A noi, per il momento, non interessano però questioni di sicurezza. Basta premere il tasto di Invio (sul tasto potrebbe esserci "↵" o Return, o Enter) per far scomparire la finestra e iniziare a lavorare.

Ad ogni icona corrisponde un programma memorizzato nella memoria secondaria, o una directory della memoria secondaria. "Cliccando" due volte, in rapida successione, su una icona si attiva il programma corrispondente (o si "apre", cioè si visualizza, la directory corrispondente). Le icone di nostro interesse per il corso sono quella etichettata con il nome "Netscape ..." e quella etichettata "TURBO C...".

Cliccare l'icona TurboC. Ora si è nell'ambiente di programmazione TurboC, visualizzato nella figura successiva.

Se si volesse ingrandire la finestra TurboC, premere *contemporaneamente* i tasti **ALT** e **Invio**.

Un ambiente di programmazione, è un software che consente (principalmente) di

1. scrivere programmi (EDITING)
2. salvare un programma scritto (registrarlo in un file in memoria secondaria)
3. aprire file contenenti programmi già scritti (per visionarne/modificarne il contenuto)
4. compilare un programma
5. eseguire un programma, interagendo (per immissione di *input* e visione di *output*) in uno *schermo di output*
6. seguire l'esecuzione di un programma, passo passo, per trovare errori (*logici, di codifica*)

Queste operazioni vengono effettuate all'interno della finestra TurboC, utilizzando i comandi contenuti nei vari *menù* (File Edit ... Help). I menù vengono aperti e chiusi e i comandi vengono selezionati ed eseguiti mediante l'uso del mouse oppure anche usando i *tasti funzione* F1 F2 ... F10 ... disponibili sulla tastiera insieme ai tasti freccia (↑↓→←) e al tasto di return (invio ...).



In figura vediamo

- la riga (o barra) dei *menù* (File Edit ... Help) in alto
- la *finestra di edit* che è quella in cui scriveremo i programmi. La finestra è contraddistinta da un nome. In mancanza di altre specifiche, il turboC chiama NONAME**. la finestra di edit. Ovviamente l'utente può decidere qual è il nome della finestra (usando il comando SAVE che vedremo dopo).
- La finestra di output (quella più in basso - numerata 1) in cui è possibile vedere l'output del programma che si è scritto e mandato in esecuzione.

Cliccando una volta in una finestra la si attiva ed appaiono le barre di *scrolling* che permettono di scorrere (appunto, *scroll*) la finestra su e giù e a destra e sinistra, cliccando sulle frecce poste agli estremi delle bande. (NB finché nella finestra non c'è nulla, lo scrolling è inattivo o poco evidente).

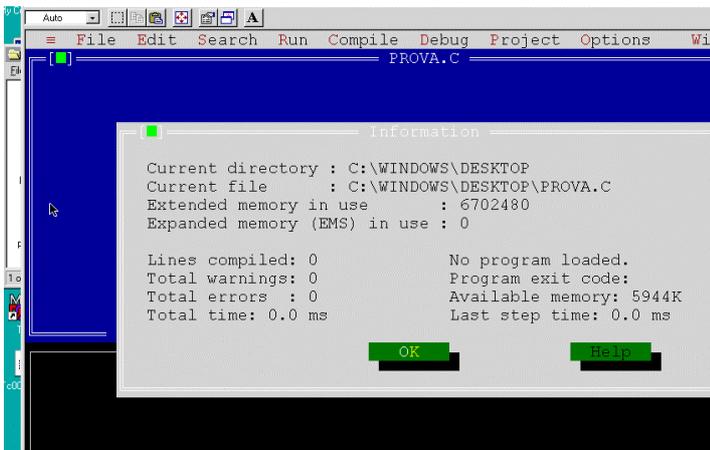
Analogamente, cliccando una volta con il mouse sul nome di un menù si apre la tendina dei comandi disponibili in quel menù. Cliccando poi sul comando corrispondente lo si esegue. (Leggere il seguito e poi mettiamo in pratica).

Se non si vuole o può usare il mouse, premendo il tasto F10 si "attiva" la barra dei menù. Usando i tasti freccia (→ ←) ci si può muovere sulla riga di menù, attivando quello che interessa premendo Invio dopo averlo selezionato. Questo fa aprire la tendina dei comandi disponibili nel menù che possono essere selezionati con i tasti freccia (↑↓) e attivati con Invio.

Esercizio: usando il mouse o i tasti (meglio se provate con entrambe le modalità) selezionare il menù **COMPILE** e in esso il comando **Information**. Si apre una finestra, chiamata *information* il cui contenuto informativo non ci interessa. Dovete però notare che in alto a appare un quadratino verde (racchiuso tra parentesi quadre) e in basso un "pulsante" con scritto OK.

Cliccando con il mouse su OK la finestra si chiude (scompare). Se non si può o vuole usare il mouse bisogna premere invio (questo "attiva" il pulsante OK).

Un'alternativa per chiudere le finestre (in generale) consiste nel cliccare sul quadratino in alto a sinistra, oppure nell'eseguire il comando **Close** contenuto nel menù **WINDOW**.



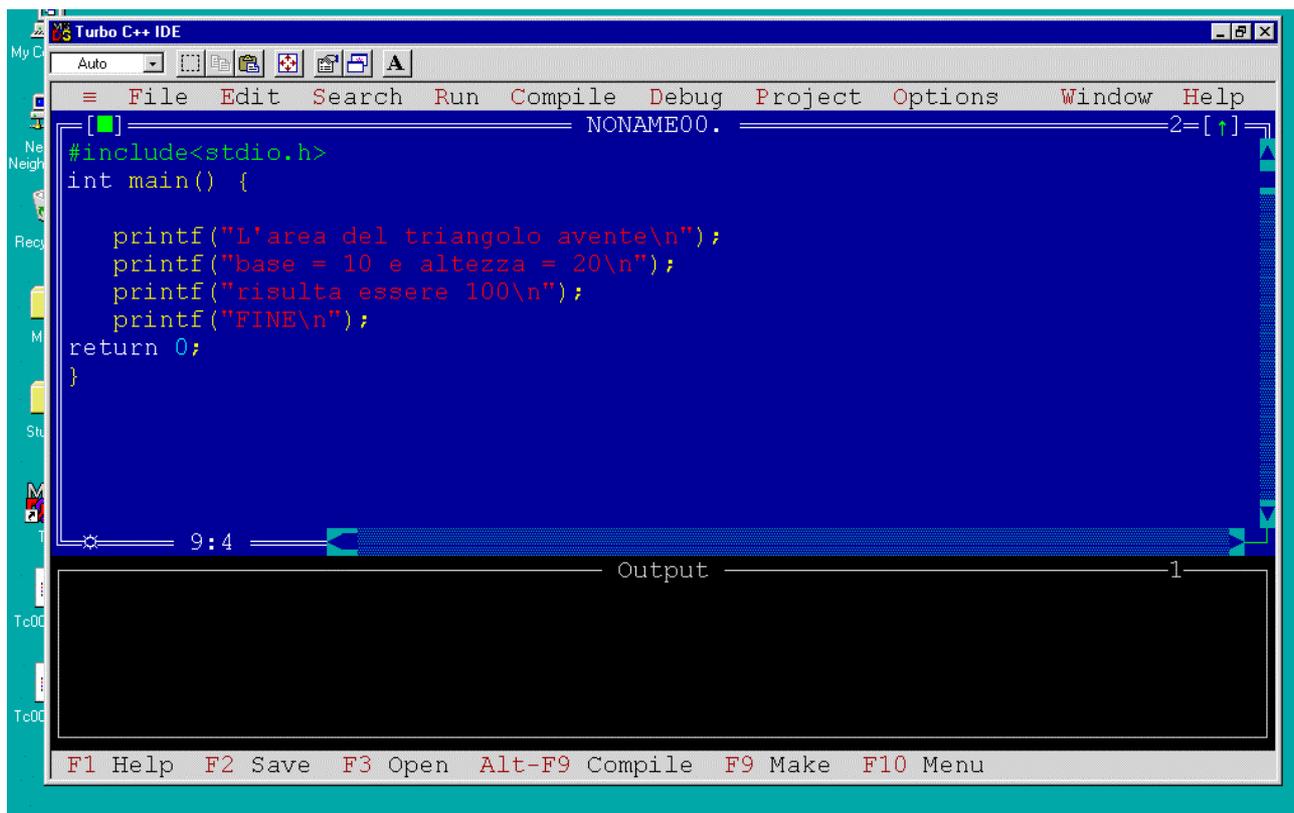
N.B. E' bene inserire ora nel driver il dischetto personale, che abbiamo portato (nell'unità base del calcolatore c'è una feritoia apposita dove inserire il dischetto). **Se** il dischetto **non** è stato ancora **formattato** è bene farlo ora, in questo modo:

accedendo alla finestra **DOS-shell** (per farlo premere F10 e poi return; si apre la tendina con il menù **FILE**; posizionarsi su *Dos-Shell* tramite le freccette ↑, ↓ e premere invio ancora: si accede ad una finestra di sistema operativo, in cui possiamo dare comandi DOS)

inviando poi il comando **FORMAT A:** Dopo aver formattato il dischetto inviare il comando **EXIT** (scrivere EXIT e premere return) per tornare in TC

1.5. Scrittura di un programma

Una volta entrati nell'ambiente TurboC, siamo pronti per impostare un primo programma. Ci si trova in una finestra intitolata "NONAME00." oppure NONAME.C, oppure NONAME.CPP. Questa è la "finestra di EDIT" in cui scriviamo il nostro programma.



```
#include<stdio.h>
int main() {

    printf("L'area del triangolo avente\n");
    printf("base = 10 e altezza = 20\n");
    printf("risulta essere 100\n");
```

```

    printf("FINE\n");
return 0;
}

```

Provate a copiare il programma, esattamente come è riportato sopra (esattamente, ossia carattere per carattere) sapendo che per correggere eventuali errori di scrittura (*editing*) potete cancellare quel che avete scritto come segue:

- posizionarvi subito prima della parte da cancellare usando
 - il mouse, muovendo la freccia/cursore sul punto e cliccando una volta
 - oppure i quattro tasti freccia ←, ↑, →, ↓
- cancellarla con il tasto Canc
- riscrivere il testo esatto.

In alternativa si può usare il tasto "backspace" (tasto largo, posizionato sopra al tasto invio ↵), posizionandosi (come fatto sopra) sul carattere successivo a quelli da cancellare.

Alcuni caratteri potrebbero non essere direttamente disponibili sulla tastiera. Bisogna sapere che

- premendo il tasto ALT
- e, tenendolo premuto, battendo sul tastierino numerico (a destra!) il codice ASCII di un carattere
- quando si rilascia ALT appare sullo schermo il carattere con quel codice ASCII

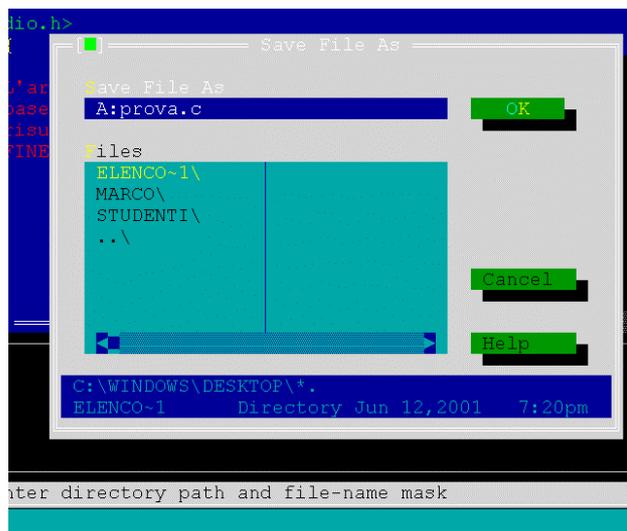
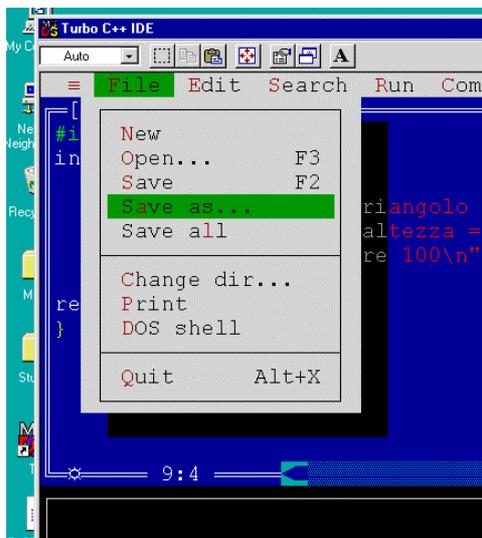
Ecco alcuni codici ASCII: {/123, }/125, [/91,] /93, #/35, | /124, ~ /126.

1.6. SALVATAGGIO DEL PROGRAMMA, COMPILAZIONE ED ESECUZIONE.

Finora abbiamo solo immesso il programma nella memoria RAM, cioè in una memoria che viene azzerata ogni volta che il calcolatore viene spento. Per trasferire il contenuto della finestra di edit nella memoria secondaria (in un file con un dato nome) dobbiamo usare il comando *Save as* del menù **FILE**. In particolare vogliamo salvare il file sul disketto (A:).

Procedere come segue

In seguito le istruzioni verranno date facendo riferimento all'uso della tastiera, ma ricordate che ogni operazione di apertura menù e selezione comandi può essere eseguita semplicemente con il MOUSE, posizionando il cursore sul punto prescelto e "cliccando".



- a) premere il tasto F10: notate che una delle parole nella parte superiore del video cambia colore; si dice che questa parola, che rappresenta uno dei menù di comandi disponibili, è stata selezionata. Usando le freccette orizzontali →, ← possiamo selezionare qualcuno degli altri menù. Selezioniamo il menù **FILE**.
- b) premere il tasto invio per "aprire" il menù **FILE**;
- c) selezionare il comando *Save as*, usando ancora i tasti freccia e premere invio;

- d) impostare un nome da assegnare al programma su disco: sarà un nome completo di un file destinato a stare sul floppy disk dell'unità A : , ad esempio A:PRIMO.PAS e premere invio .

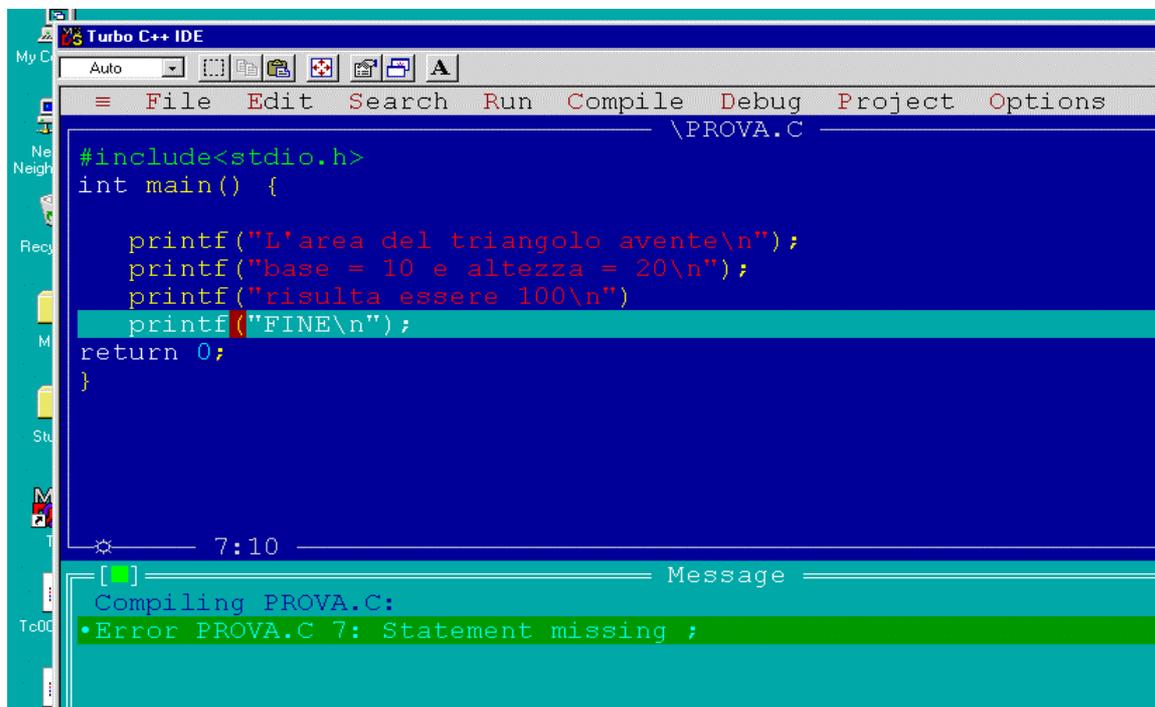
Ricordare che se, dopo qualche correzione, vorremo salvare di nuovo il programma nel file PRIMO.PAS del disco A:, basterà eseguire il comando *Save* (che si applica a file che hanno già un nome, diverso da NONAME**, e che non prevede la scrittura esplicita del nome del file).

Compilazione del programma visualizzato nella finestra di EDIT.

- premere il tasto F10 e usare i tasti freccia ←, → per selezionare il menù **COMPILE**;
- premere il tasto invio per “aprire” il menù e poi selezionare il comando *Compile* (stesso nome del menù), usando ancora i tasti freccia e premere invio;

Se il programma è corretto appare il messaggio “Compile successful”, altrimenti può comparire un messaggio di errore (in rosso), e il cursore si posiziona sull’istruzione che ha causato l’errore, o su una delle successive. Quando compare un messaggio di errore, se si vogliono maggiori spiegazioni su tale errore si può premere F1. Dopo bisogna premere ESC per tornare nella finestra di edit (cioè sul testo del programma)

Un esempio di errore: manca un ';' in un'istruzione. Il processo di compilazione si e' bloccato



```
#include<stdio.h>
int main() {

    printf("L'area del triangolo avente\n");
    printf("base = 10 e altezza = 20\n");
    printf("risulta essere 100\n");
    printf("FINE\n");
return 0;
}
```

7:10

Message

Compiling PROVA.C:
•Error PROVA.C 7: Statement missing ;

sull'istruzione successiva:

Quando la compilazione ha avuto successo, si può eseguire il programma:

Esecuzione del programma visualizzato nella finestra di edit

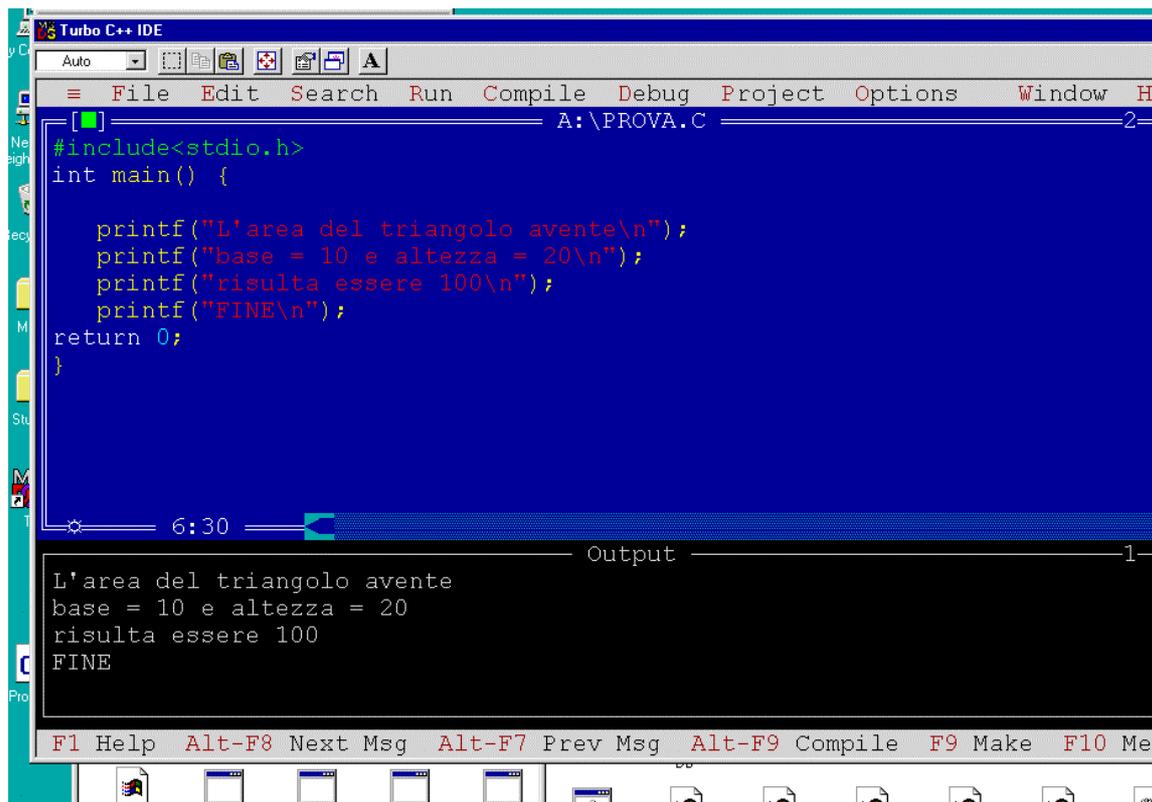
Procedere come segue:

- premere il tasto F10 e selezionare il menù **RUN**;
- premere il tasto invio per “aprire” il menù, selezionare il comando *Run*, e premere invio;

Il programma stampa in output quel che deve stampare. Dopo l'esecuzione si torna subito sulla finestra di edit. , ma l'output del programma e' mostrato nella finestra di output. Se la *finestra di output* non era (è) visibile la si può attivare eseguendo il comando *Output* del menù **WINDOWS**.

- premere il tasto F10; usare i tasti freccia per selezionare il menù **WINDOWS**
- premere il tasto invio per “aprire” il menù; selezionare *Output*, e premere invio: apparirà lo schermo con i dati di input/output del programma.

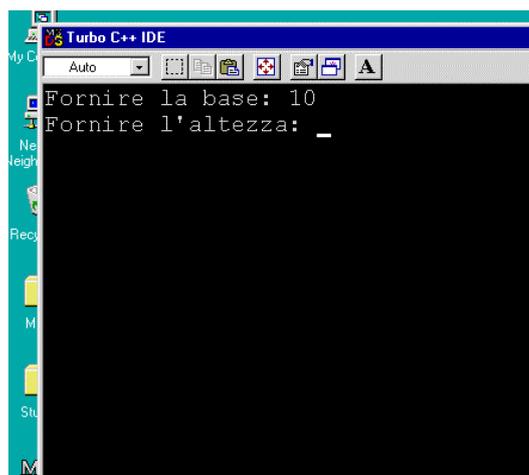
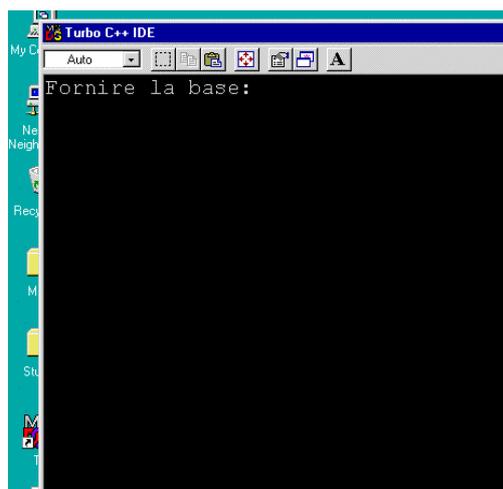
La finestra può essere "scrollata" per vederne il contenuto.



Esercizio: modificare il programma `prova.c` in modo da usare le variabili `base` e `altezza`, come fatto a lezione: il programma legge le due variabili da input e calcola l'area stampando l'output.

Una volta aggiunte le istruzioni necessarie, salvare (**Save**) compilare ed eseguire una volta.

Vedi figure successive. L'esecuzione del programma appare a tutto schermo, su quello che si chiama "User Screen". Viene richiesto l'input della base e poi dell'altezza. Ad ogni input fornito da tastiera bisogna premere Invio.



Lo User Screen rimane visibile solo durante l'esecuzione del programma, perciò nel nostro esempio scompare appena finita l'esecuzione del programma (facendo riapparire la finestra di edit).

Per visualizzare lo User Screen anche al di fuori dell'esecuzione si deve selezionare il comando **User Screen** dal menù **WINDOWS**:

- premere il tasto F10; usare i tasti freccia per selezionare il menù **WINDOWS**;

- premere il tasto invio per “aprire” il menù; selezionare *User Screen*, e premere invio: apparirà lo schermo con i dati di input/output del programma.
Poi si torna all'ambiente TP premendo Invio

Quando ci sono più finestre aperte sullo schermo, il loro posizionamento e la loro visibilità sono governati mediante i comandi del menù **WINDOWS**. In particolare, i comandi *Next* e *Previous* permettono di "rendere attive" una o l'altra delle finestre già aperte

Altrimenti si può semplicemente cliccare su una finestra per attivarla ...

Provate ad eseguire il programma per diversi dati di input e a valutare gli output conseguenti.

1.7. USCITA DAL TURBOC

Terminato il lavoro potete utilizzare di nuovo il menù **FILE** per uscire dall'ambiente Turbo Pascal: è sufficiente selezionare il comando *Exit*. Se il programma che si trova nella finestra di edit è stato modificato dopo l'ultima volta che l'avete salvato, vi verrà richiesto se volete salvare (premendo il tasto Y) o no (premendo il tasto N) le ultime modifiche.

1.8. RIENTRO E RIESECUZIONE DEL PROGRAMMA.

È possibile a questo punto rientrare nel Turbo C, secondo le modalità spiegate nella sezione 1.4, e caricare di nuovo lo stesso programma dal disco, per eseguirlo nuovamente o modificarlo: nel menù **FILE** è presente il comando *Open* che serve a questo scopo. Ovviamente è necessario scrivere, nell'apposito spazio dell'apposita finestra, il nome assegnato precedentemente al programma (cioè il nome del file in cui il programma era stato salvato con il comando *Save*). La finestra di Open appare quando il comando Open è stato selezionato.

1.9. ESERCIZIO (*capital1.c, capital2.c*).

Seguendo le modalità sperimentate nelle sezioni precedenti, scrivere un programma che, ricevuto in input il capitale iniziale di un investimento (double), stampi i valori che il capitale assume nei tre anni seguenti, supponendo che si rivaluti al 10% annuo.

Soluzioni proposte in `capital1.c` e `capital2.c`.

1.10. ESERCIZIO (*moquette.c*).

Seguendo le modalità sperimentate nelle sezioni precedenti, scrivere un programma che, ricevute in input le dimensioni (larghezza, lunghezza, in metri) di una stanza e sapendo che il costo di un mq. di moquette costa Euro 10.12, calcoli e stampi il costo per la copertura in moquette della stanza.

Soluzione proposta in `moquette.c`.