

COSTRUIRE UNA APPLICAZIONE IN C

- Per “costruire” un’applicazione occorre:
 1. Compilare il file (o i file se più d’uno) che contengono il testo del programma (*file sorgente*)

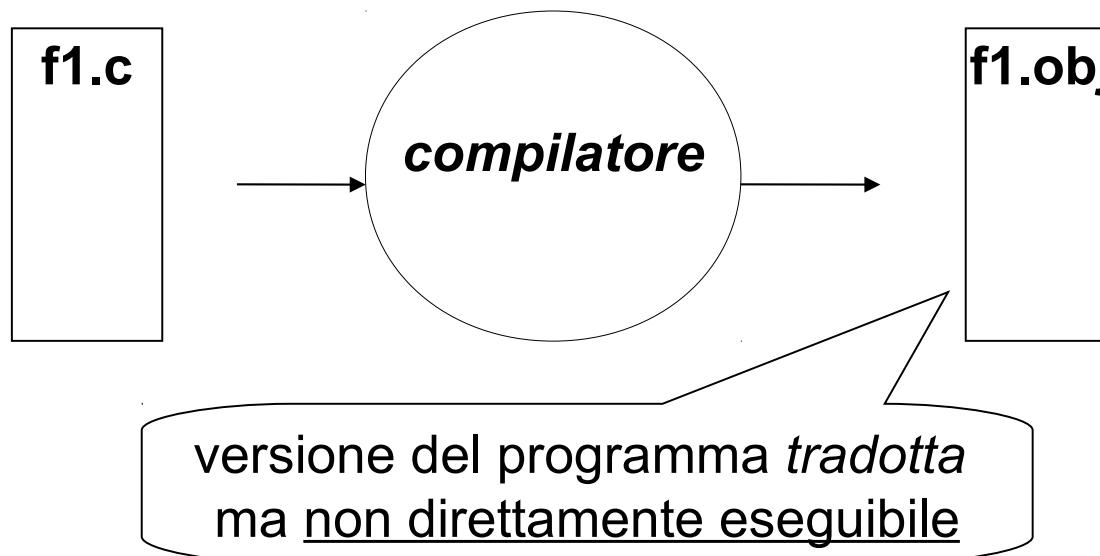
Il risultato sono uno o più file *oggetto*.

1. Collegare i file oggetto l’uno con l’altro *e con le librerie di sistema.*

COMPILARE (COMPILE)

1) Compilare il file (o i file se più d'uno)

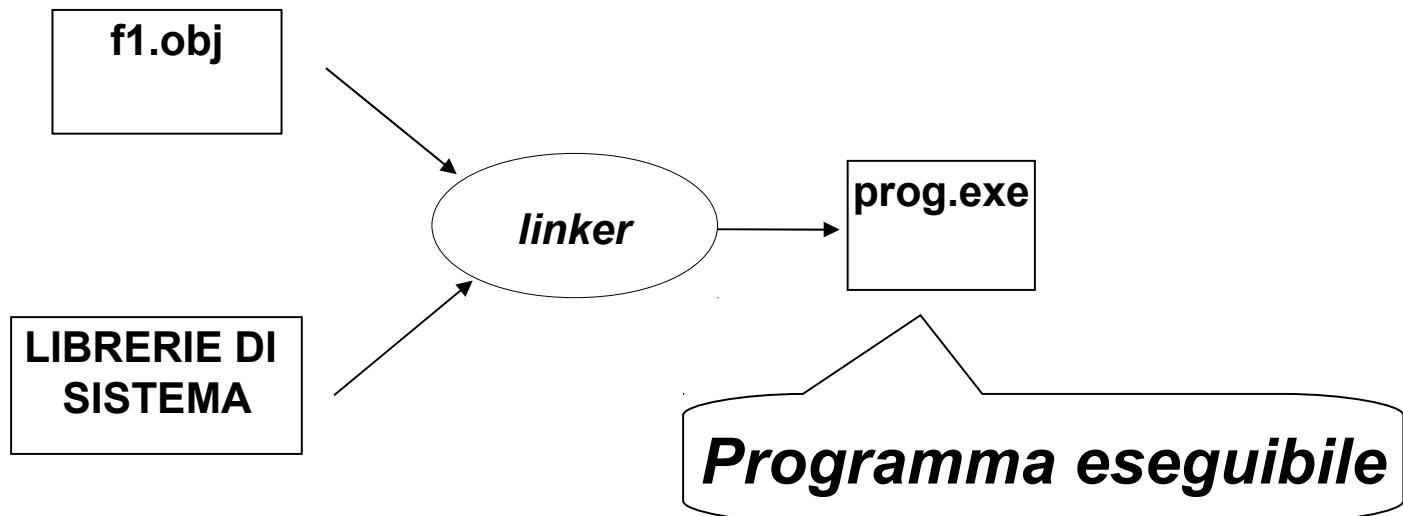
- File sorgente: estensione **.c**
 - File oggetto: estensione **.o** o **.obj**



COLLEGARE (LINK)

2) Collegare il file (o i file) oggetto fra loro e *con le librerie di sistema*

- File oggetto: estensione **.o** o **.obj**
- File eseguibile: estensione **.exe** o nessuna



LE LIBRERIE DI SISTEMA

- Le librerie di sistema sono componenti software che consentono di:
 - interfacciarsi col sistema operativo
 - usare le risorse da esso gestite e
 - realizzare alcune "istruzioni complesse" del linguaggio

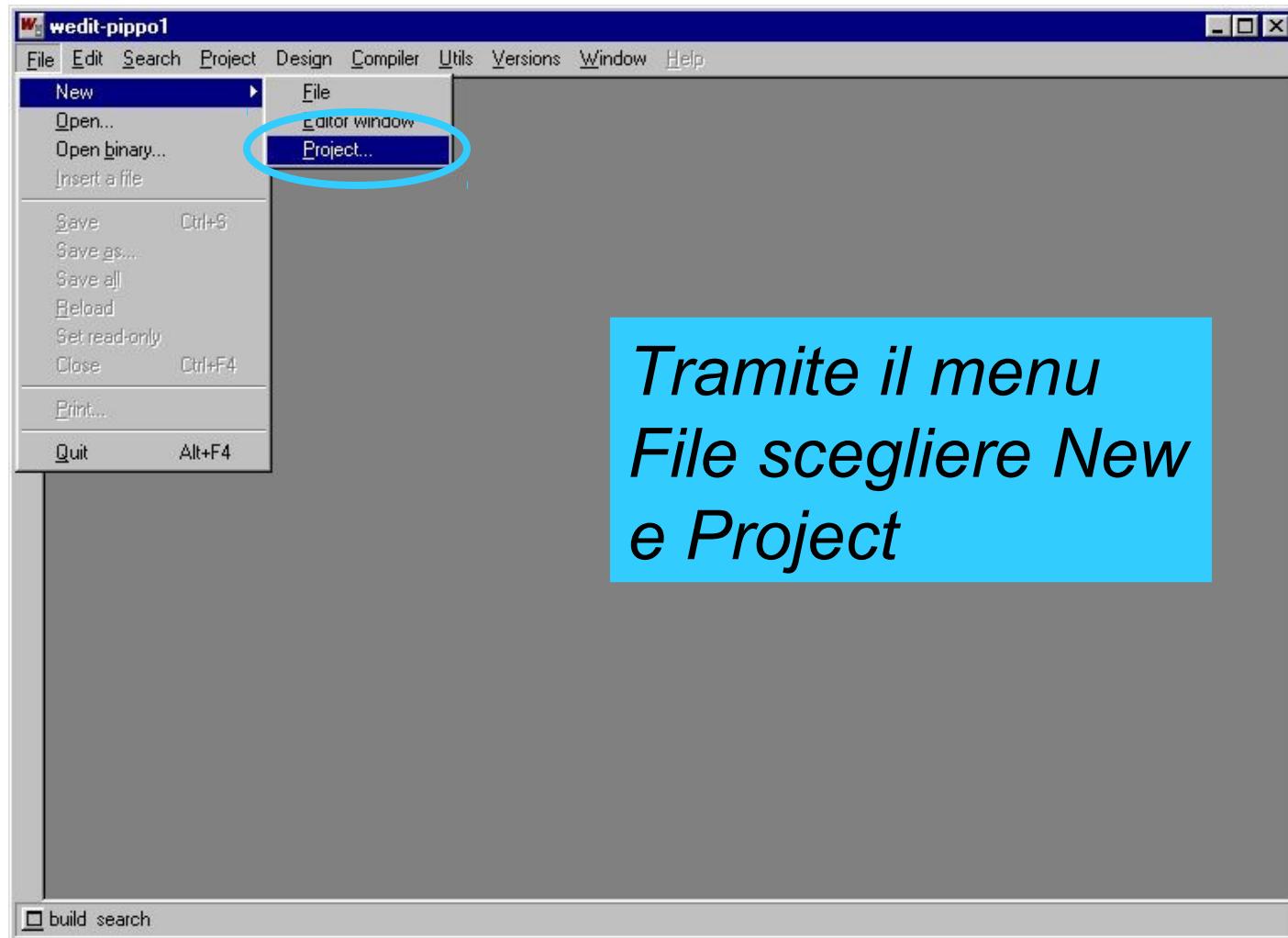
AMBIENTI INTEGRATI (IDE)

- Oggi, gli *ambienti di lavoro integrati* automatizzano la procedura:
 - compilano i file sorgente (*se e quando necessario*)
 - invocano il linker per costruire l'eseguibile
- Per fare ciò, devono sapere:
 - quali file sorgente costituiscono l'applicazione
 - il nome dell'eseguibile da produrre.

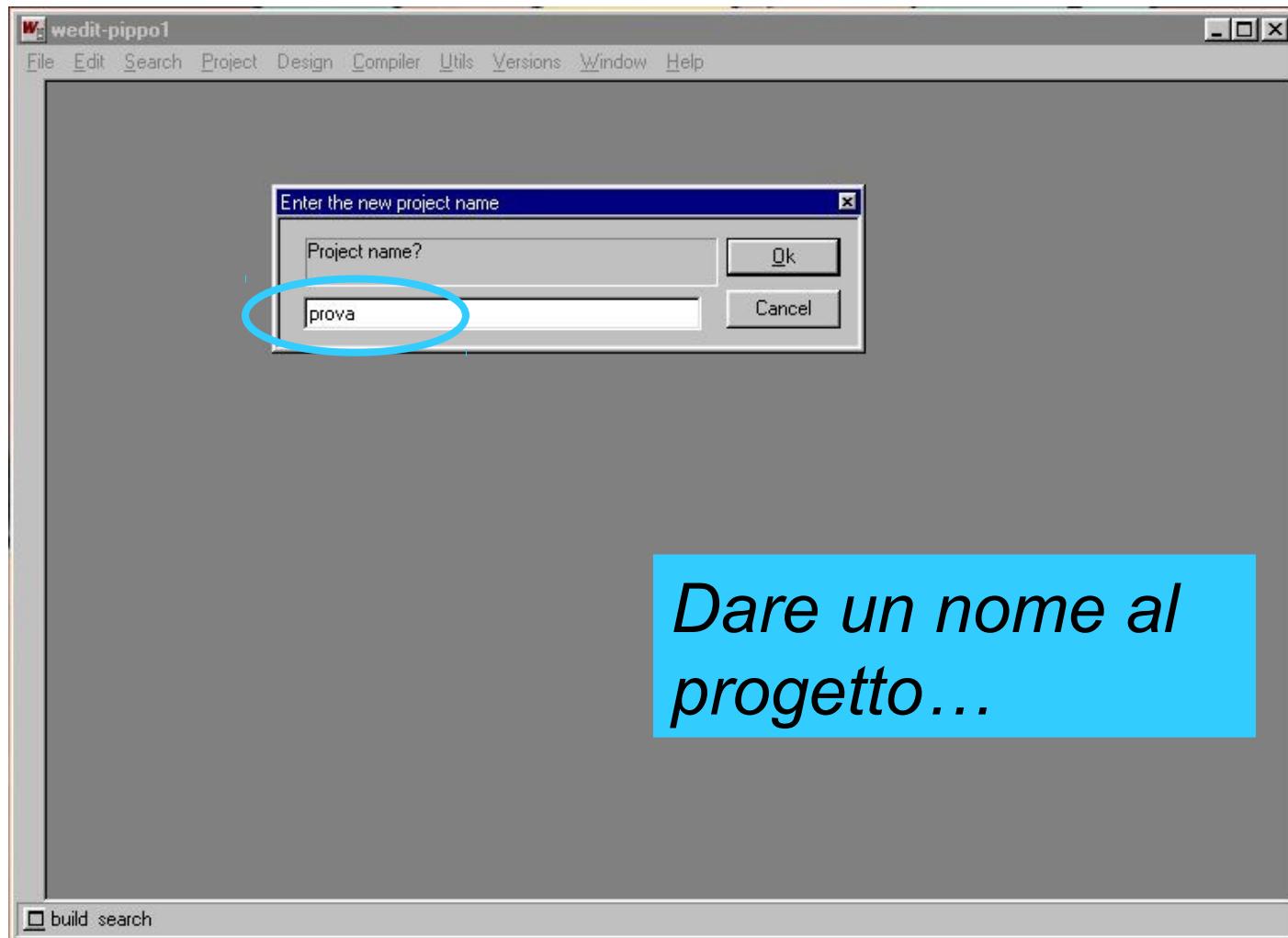
IL PROGETTO

- Da queste esigenze nasce il concetto di **progetto**
- Un progetto è
 - un *contenitore* concettuale (e fisico)
 - che elenca i file sorgente in cui l'applicazione è strutturata
 - ed eventualmente altre informazioni utili.
- Oggi tutti gli IDE supportano il concetto di progetto
 - per qualunque linguaggio
 - e sistema operativo/ambiente

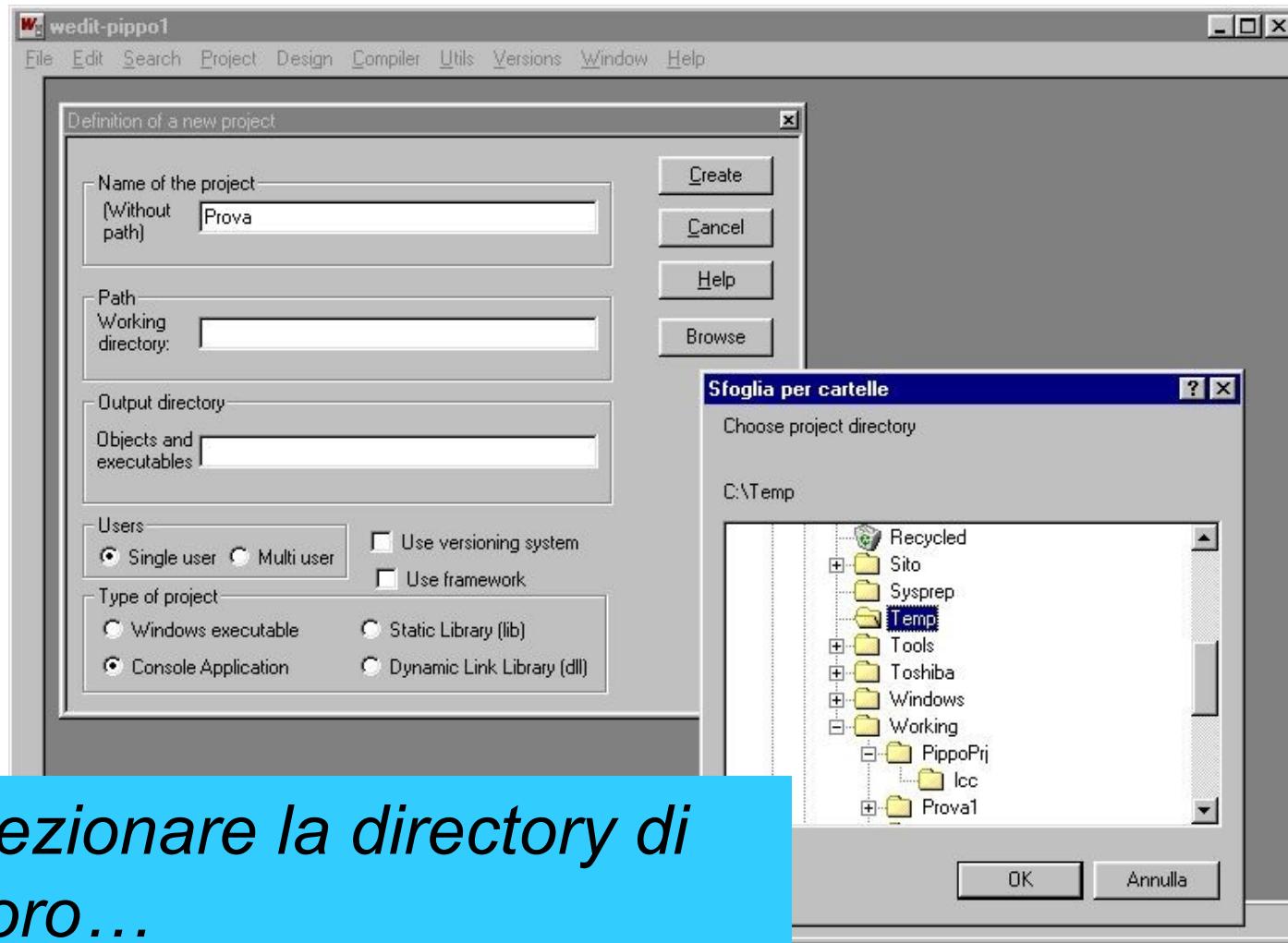
CREARE UN PROGETTO IN LCC (1)



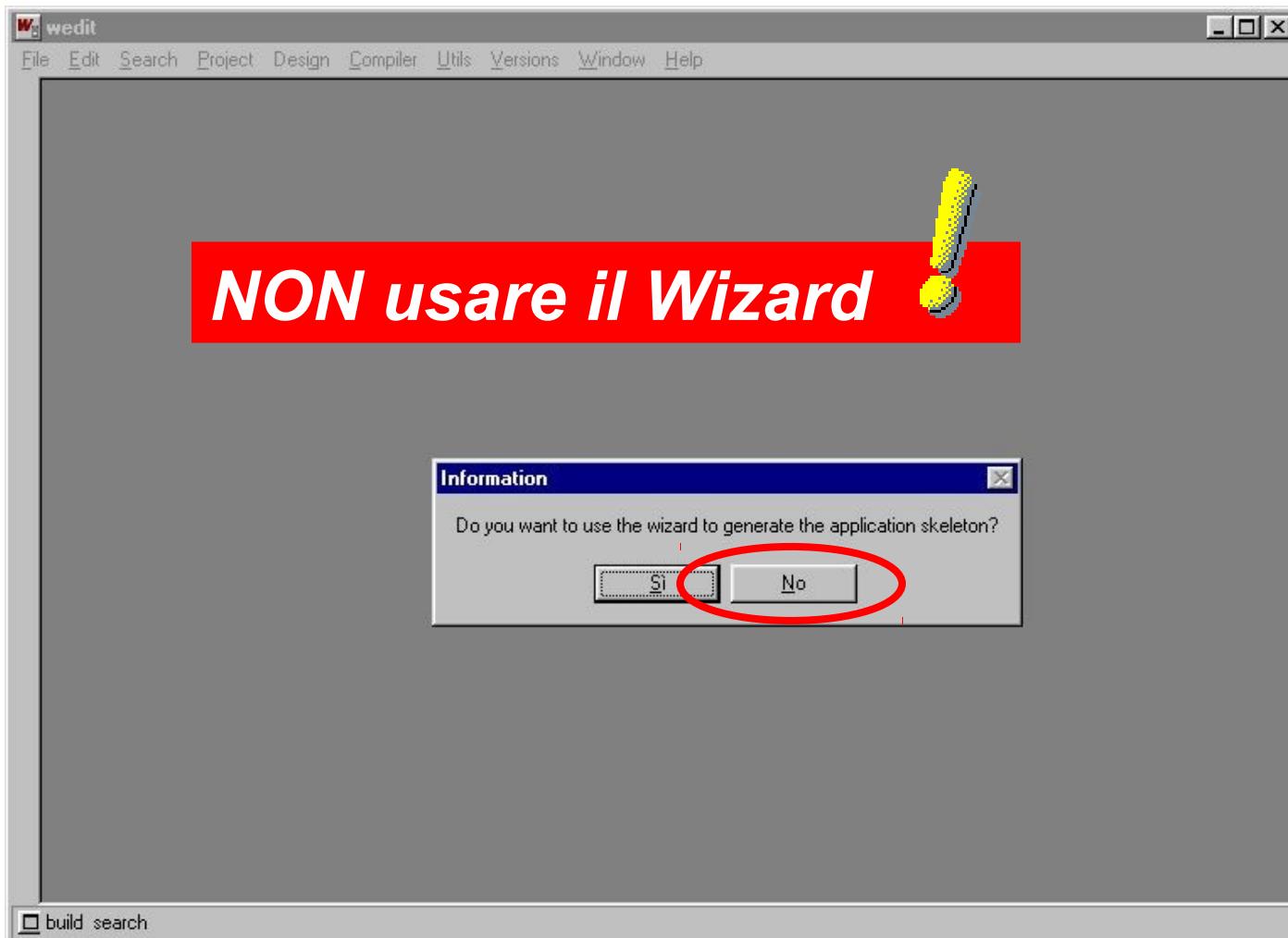
CREARE UN PROGETTO IN LCC (2)



CREARE UN PROGETTO IN LCC (3)



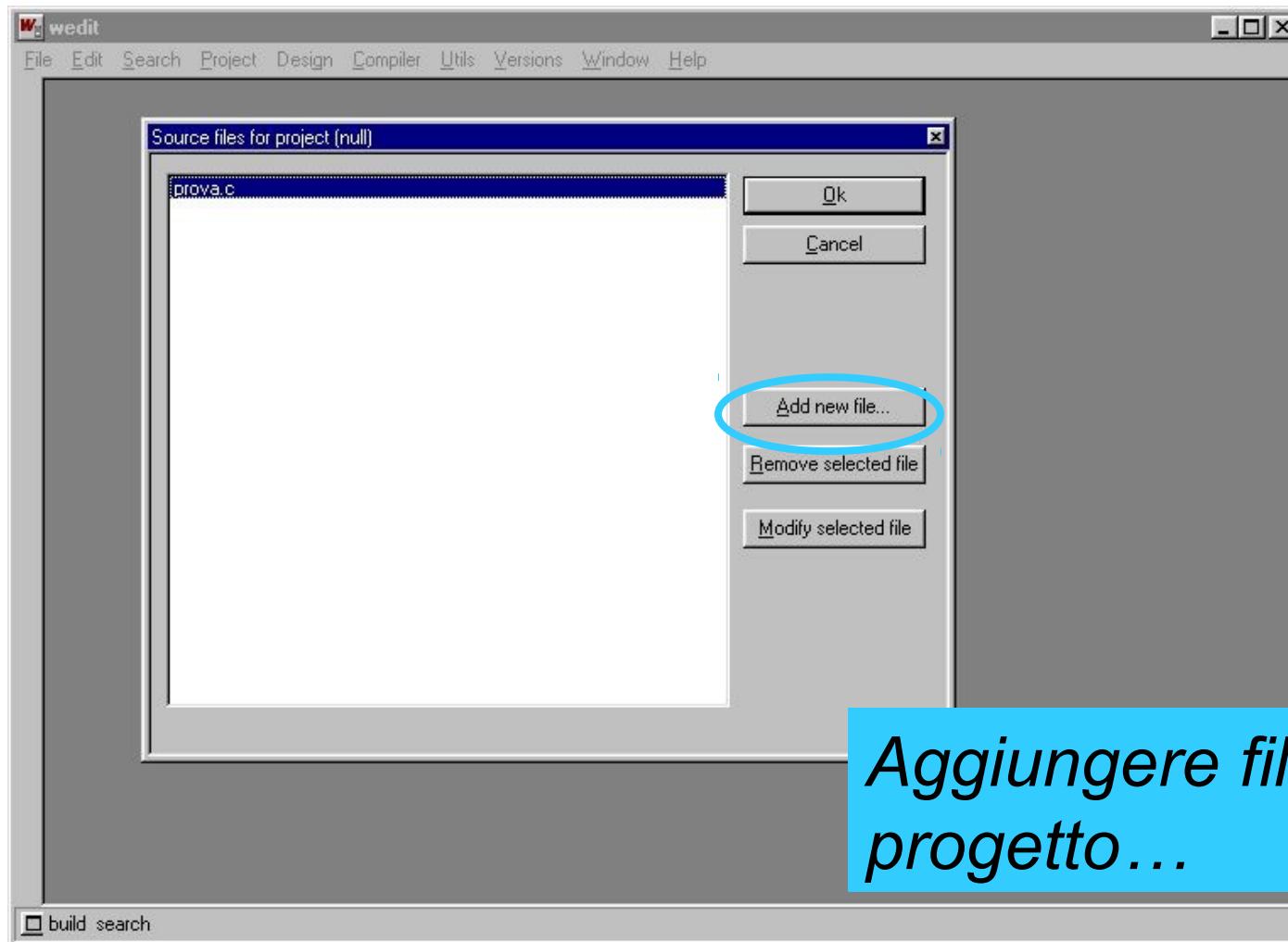
WARNING



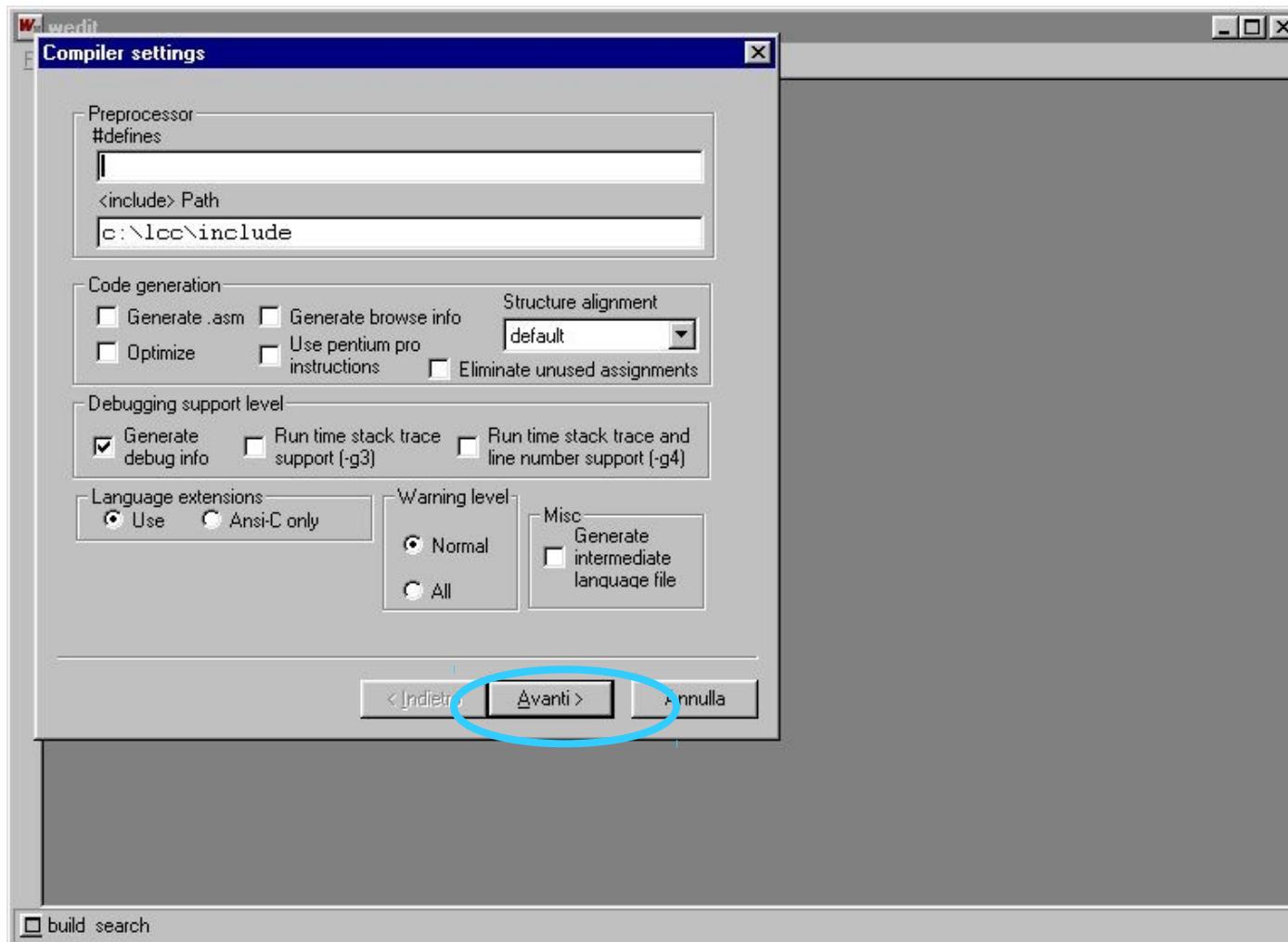
AGGIUNGERE FILE AL PROGETTO



AGGIUNGERE FILE AL PROGETTO (2)



AGGIUNGERE FILE AL PROGETTO (3)



IL PRIMO PROGRAMMA C

```
#include <stdio.h>

/* programma che stampa a video la famosa frase
"Hello World!" */

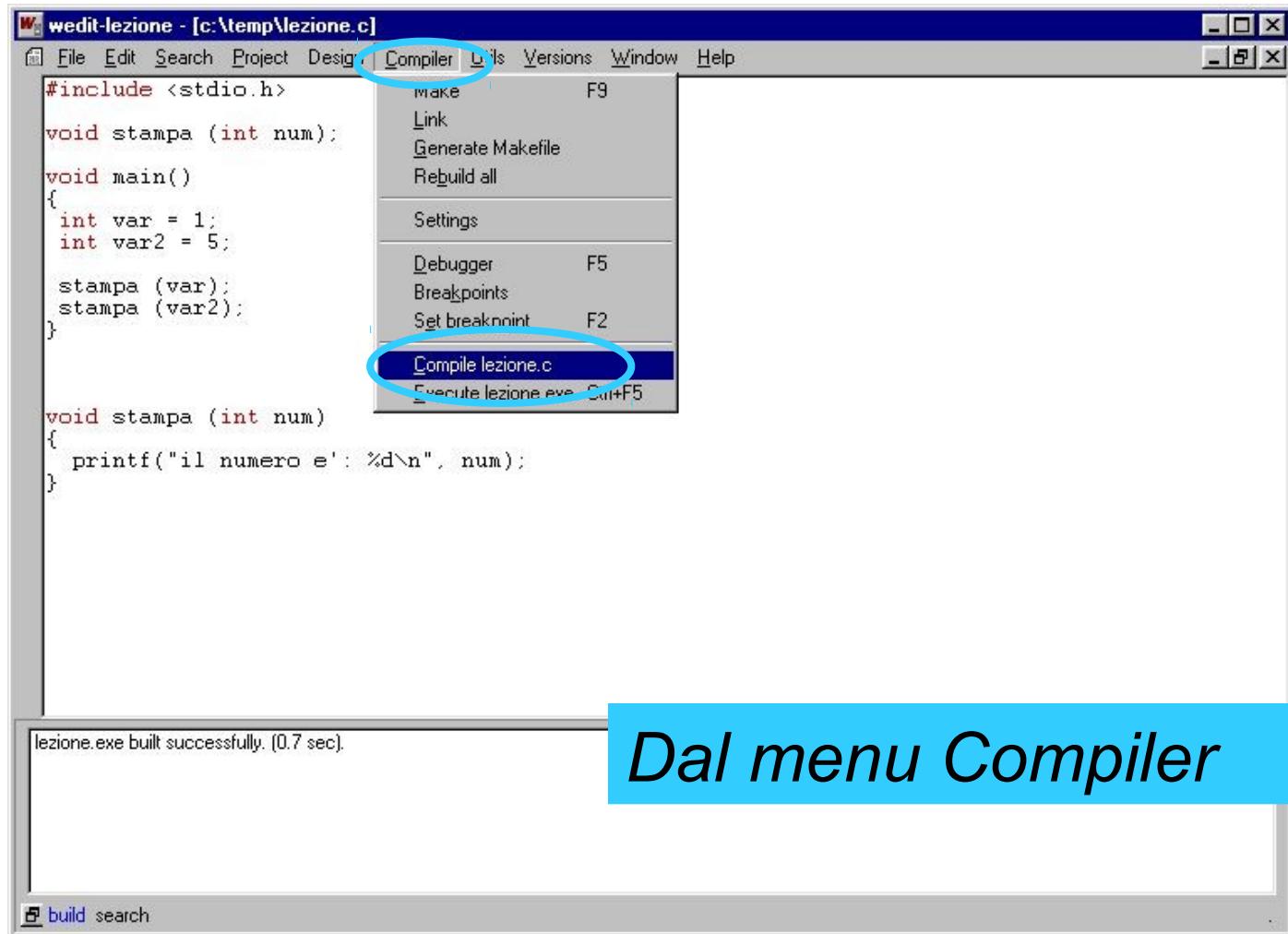
main() {
    /* parte dichiarativa vuota */
    printf("Hello World!\n"); /* p.istruzioni*/
    printf("Questo programma non fa grandi
cose...");

}
```

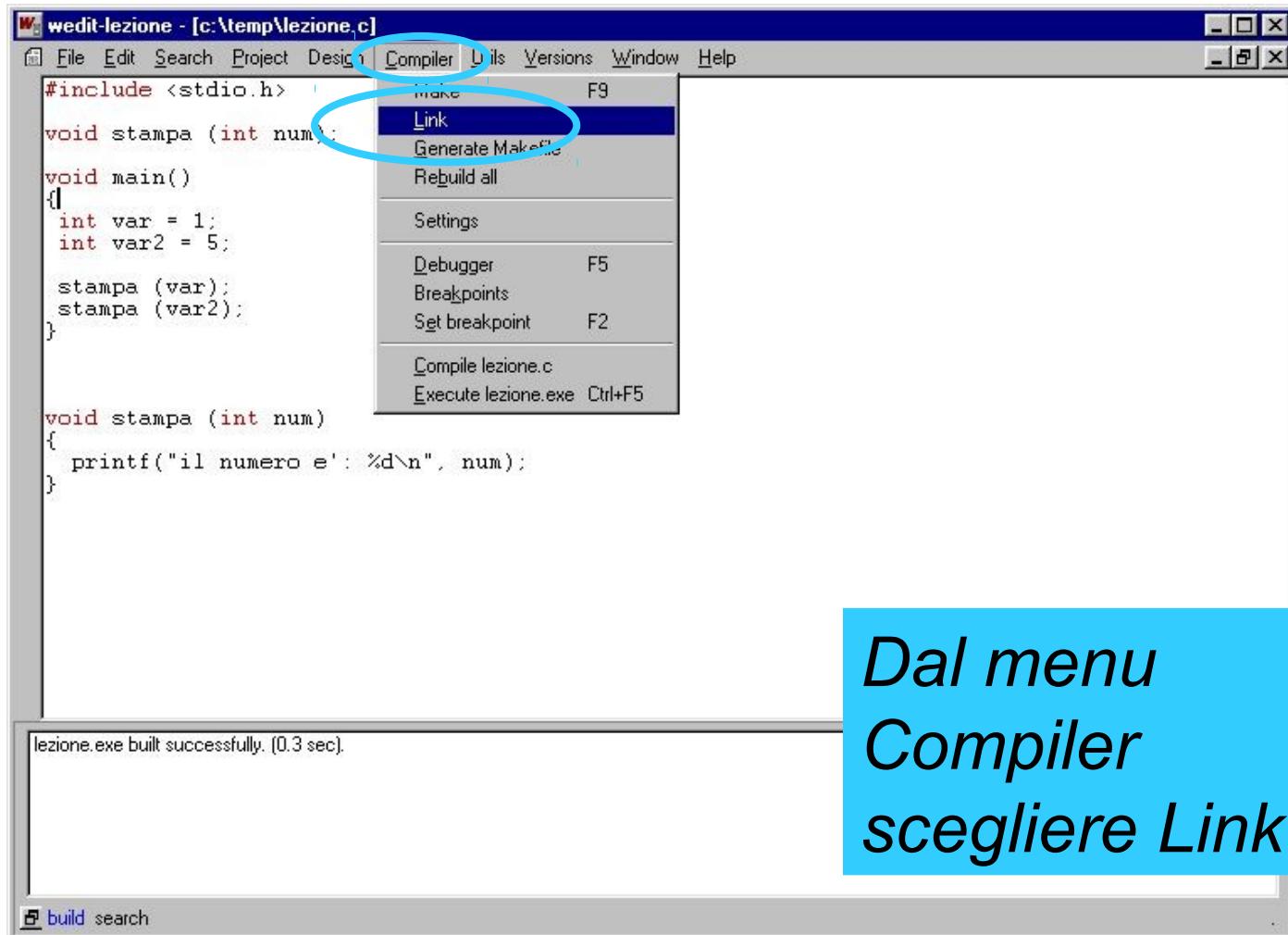
EDITARE E SALVARE



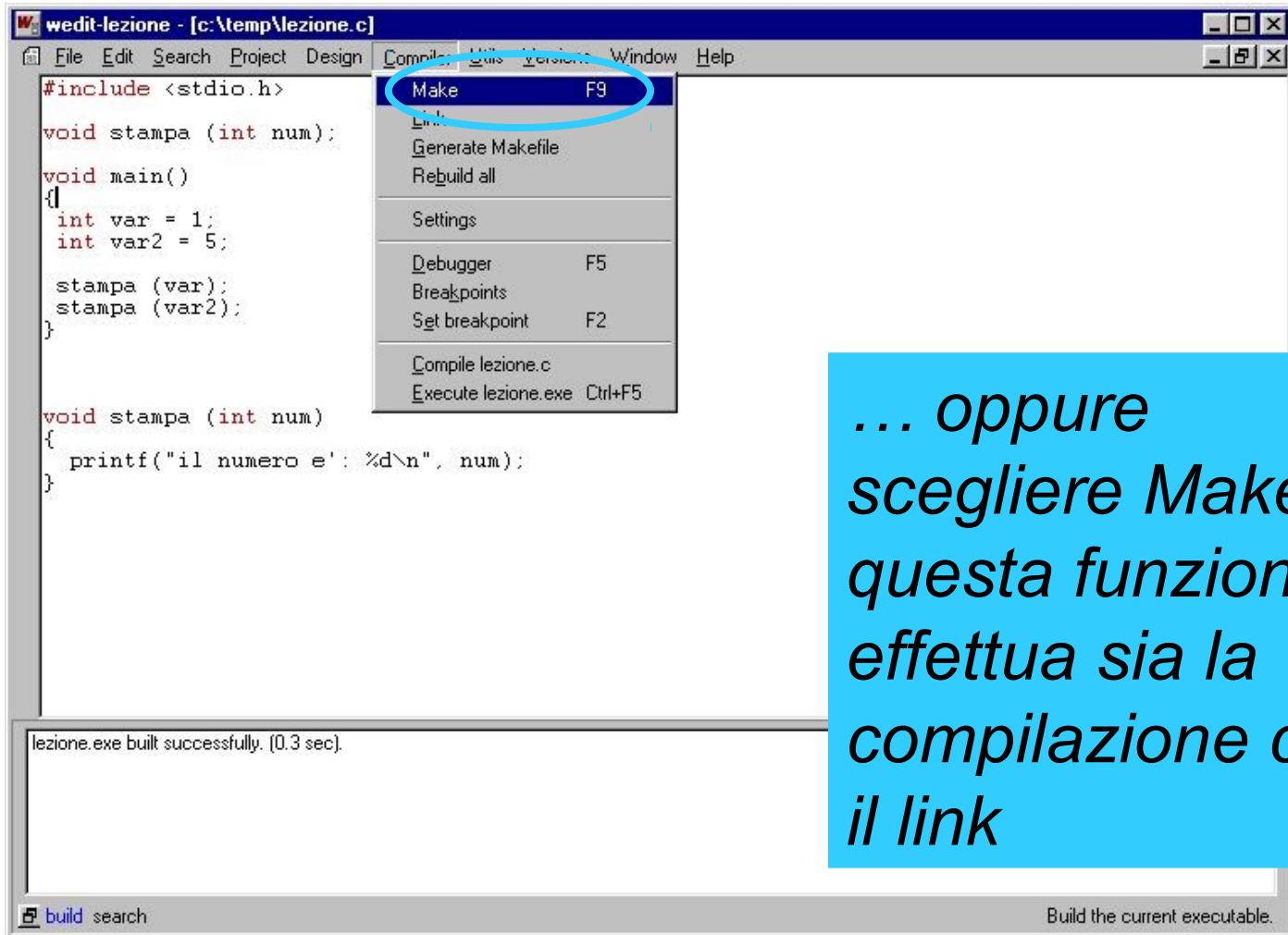
COMPILARE



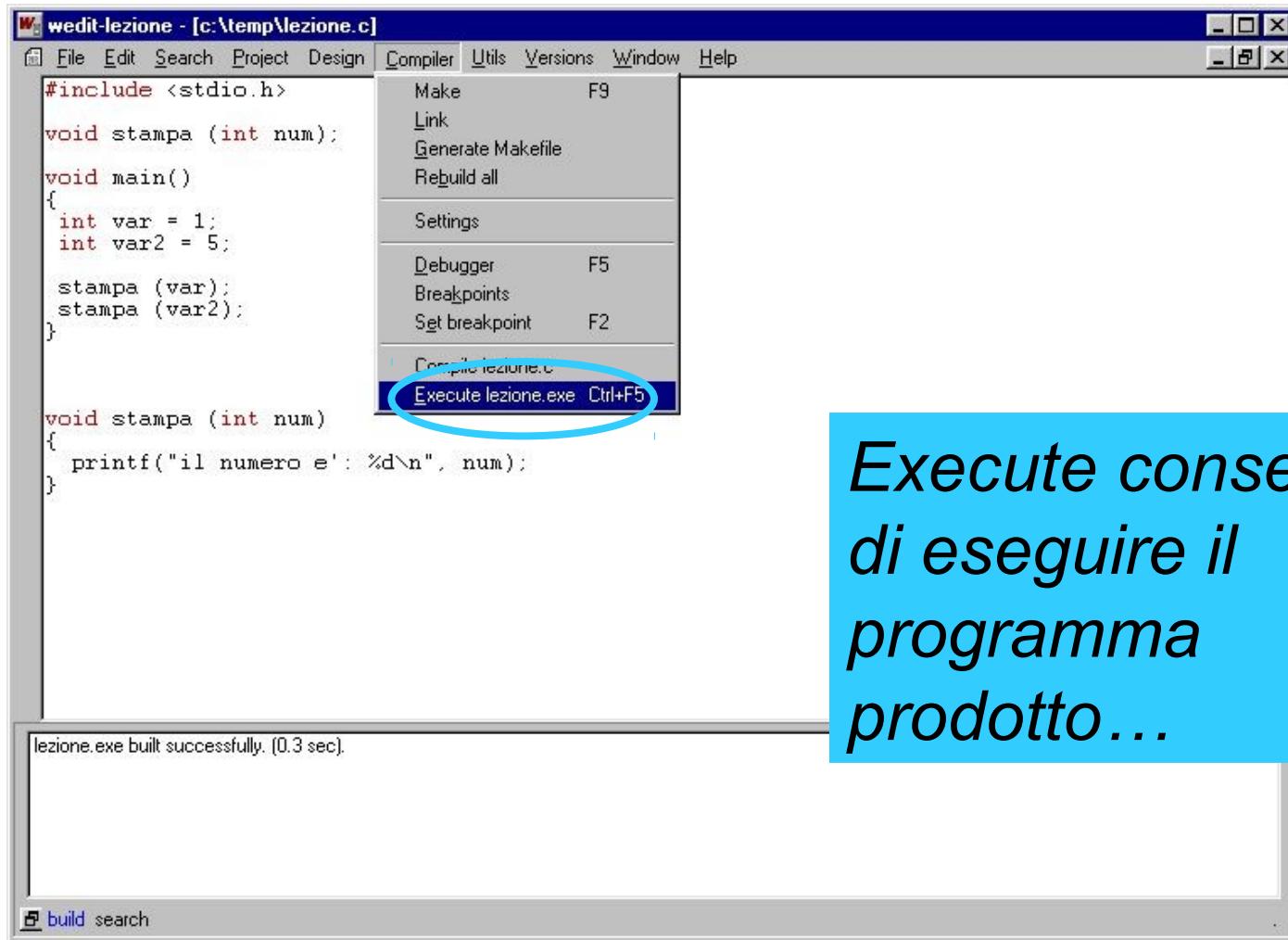
LINK



MAKE



EXECUTE



Execute consente di eseguire il programma prodotto...

EXECUTE

*... e visualizza
l'output*

The screenshot shows a Windows application window titled "lezione". The window has a menu bar with "File", "Edit", "View", "File", "Window", and "Help". Below the menu is a toolbar with various icons. The main area of the window contains the output of a C program:

```
int var = 1;
int var2 = 5;
stampa();
stampa();
}
il numero e': 1
il numero e': 5

void stampa("c:\\temp\\lcc\\lezione.exe"
{
    printf(
}
"Return code 16
Execution time 0.011 seconds
Press any key to continue...
```

At the bottom left of the window, there is a status bar with the text "lezione.exe built". At the bottom right, it says "stampa 19:2".

IL SECONDO PROGRAMMA C

```
#include <stdio.h>

/* programma che, letti due numeri a terminale,
ne stampa la somma*/

main() {
    int X,Y;      /* p. dichiarativa */
    scanf("%d%d",&X,&Y);      /*p.istruzioni*/
    printf("%d",X+Y);
}
```