

# **IDE DevC++ 4.9.8.1.0**

## **Manuale utente**

Data ultima revisione: 22/01/2005

**Informatica**

Autori: Prof. Domenico BENEVENTANO  
Ing. Massimo GHIDONI  
Dott.sa Marina MOSCATELLI

---

## Sommario

<b>1. INTRODUZIONE .....</b>	<b>3</b>
<b>2. SETUP .....</b>	<b>5</b>
<b>2.1. Procedura di installazione .....</b>	<b>5</b>
<b>2.2. Configurazione .....</b>	<b>9</b>
2.2.1. Opzioni di compilazione .....	10
2.2.2. Opzioni dell'editor .....	11
<b>3. SESSIONE DI LAVORO .....</b>	<b>12</b>
<b>3.1. Il progetto .....</b>	<b>12</b>
3.1.1. Creazione di un nuovo progetto.....	12
3.1.2. Apertura di un progetto/file esistente.....	14
3.1.3. Modifica del codice sorgente .....	15
<b>3.2. Esecuzione del programma .....</b>	<b>16</b>
3.2.1. Compilazione .....	16
3.2.2. Compilazione, link ed esecuzione.....	19
3.2.3. Esecuzione con parametri .....	22
<b>3.3. Salvataggio del codice .....</b>	<b>23</b>
<b>4. ALCUNI RIFERIMENTI UTILI.....</b>	<b>25</b>
<b>4.1. Help in linea .....</b>	<b>25</b>
<b>4.2. Bloodshed.net.....</b>	<b>25</b>
<b>4.3. SourceForge.NET .....</b>	<b>25</b>
<b>4.4. Forum di sviluppatori .....</b>	<b>26</b>

## 1. Introduzione

DevC++ è un ambiente di sviluppo integrato (IDE, ovvero Integrated Development Environment) che funziona da front-end verso il compilatore GCC che è pienamente compatibile con lo standard C89 o ANSI C.



Fig. 1.1: Informazioni su DevC++

ANSI C è uno standard proprio perchè la maggior parte dei compilatori soddisfa le specifiche da esso previste per il linguaggio.

Grazie alla sua interfaccia grafica, un ambiente IDE consente di utilizzare in modo semplice gli strumenti offerti dal compilatore senza dover ricorrere necessariamente alle istruzioni da riga di comando (command-line).

L'ultima versione ad oggi disponibile è la 4.9.8.10, che si ottiene installando l'aggiornamento della versione 4.9.8.0.

L'IDE della 4.9.8.10 si presenta come in fig. 1.2.

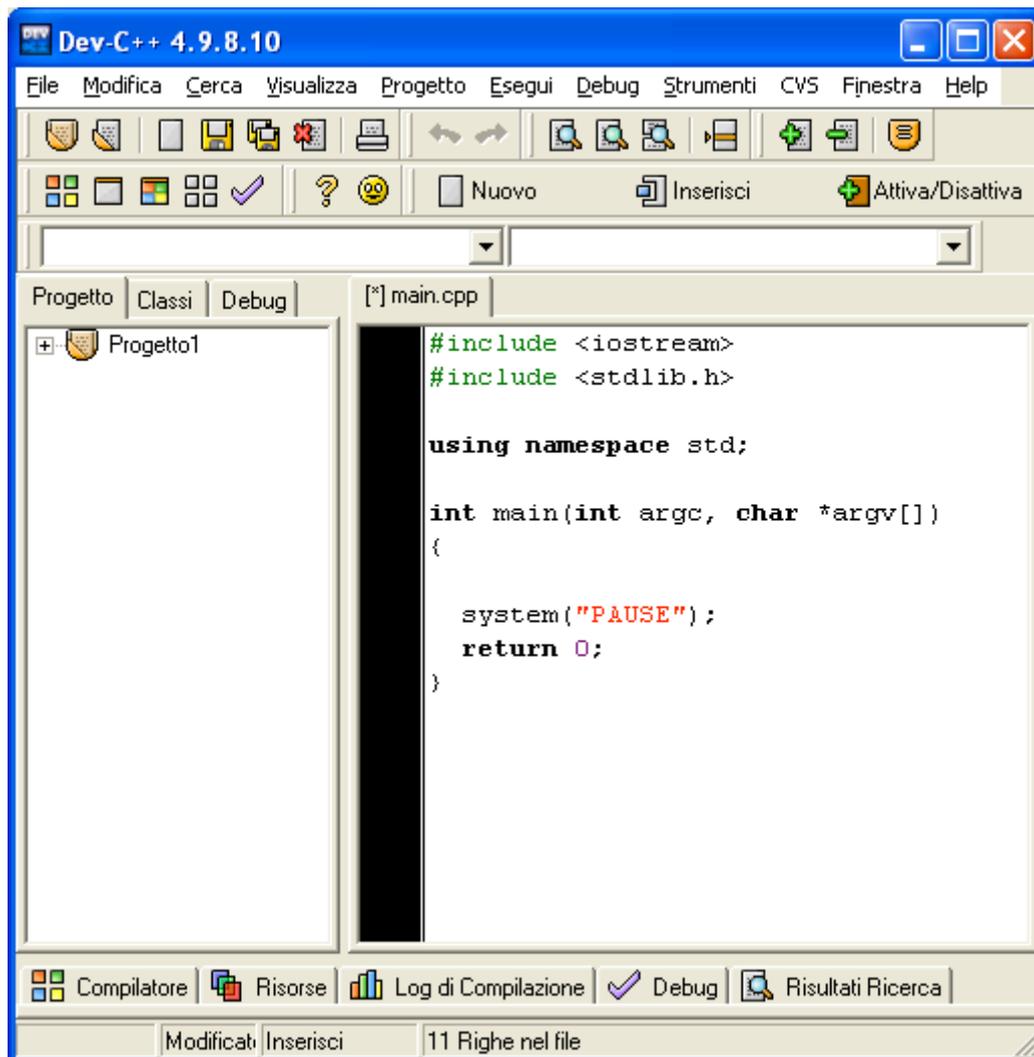


Fig. 1.2: L'IDE DevC++ 4.9.8.10

## 2. Setup

Per installare e configurare DevC++ 4.9.8.10 è necessario:

1. installare DevC++ versione 4.9.8.0 mediante doppio clic sul file *devcpp4980.exe*;
2. installare l'aggiornamento versione 4.9.8.10 mediante doppio clic sul file *devcpp49810.exe*;
3. configurare DevC++ per prepararlo all'utilizzo.

### 2.1. Procedura di installazione

Prevede i passi di seguito riportati (wizard).

Dapprima il licence agreement: DevC++ è gratuito.

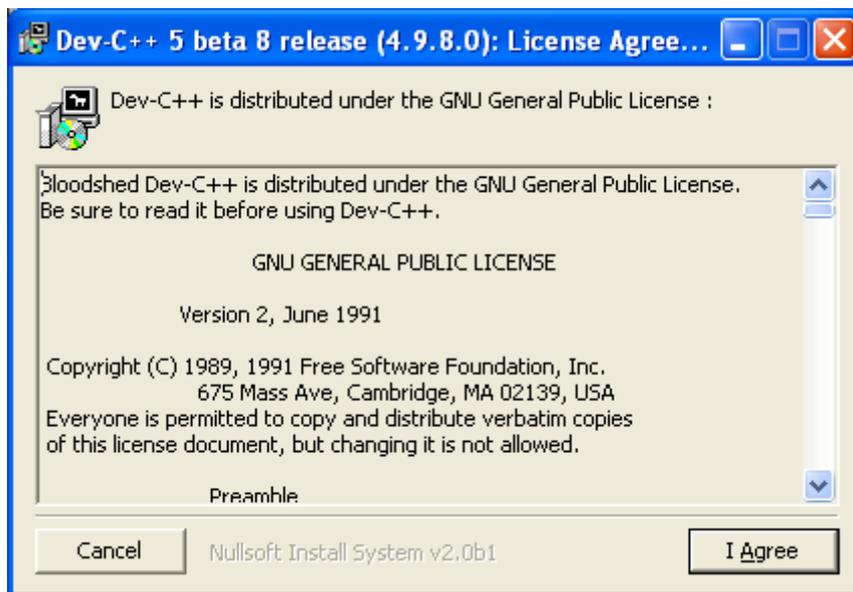


Fig. 2.1: *License Agreement*

Poi le opzioni di installazione: l'installazione tipica è di default e prevede la selezione automatica dei componenti necessari.

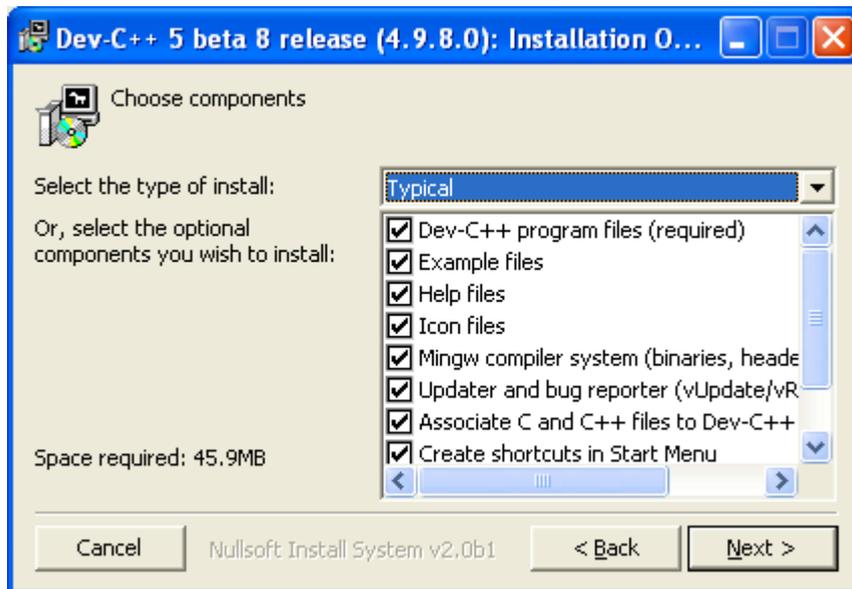


Fig. 2.2: *Installazione tipica*

Si sceglie quindi la directory di installazione.

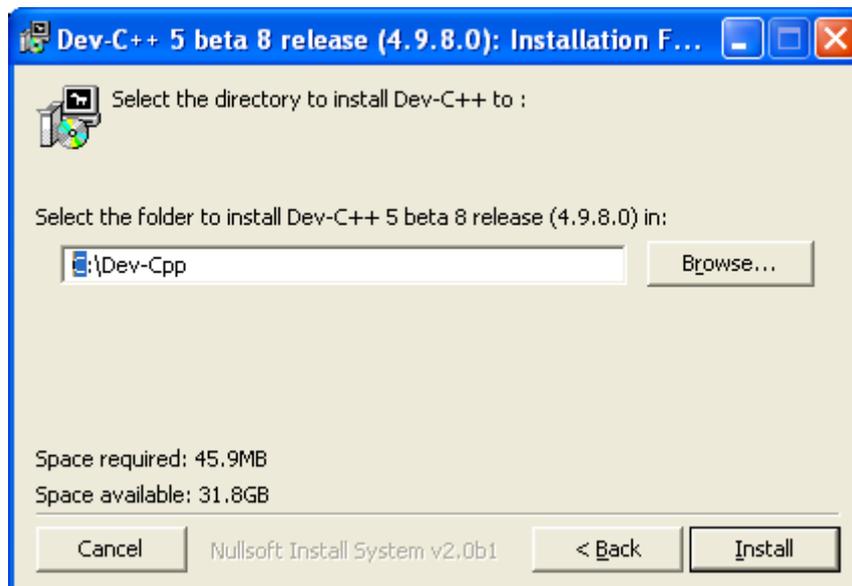


Fig. 2.3: *Directory di installazione*

Vengono estratti nella directory specificata i file compressi contenuti del file di installazione.

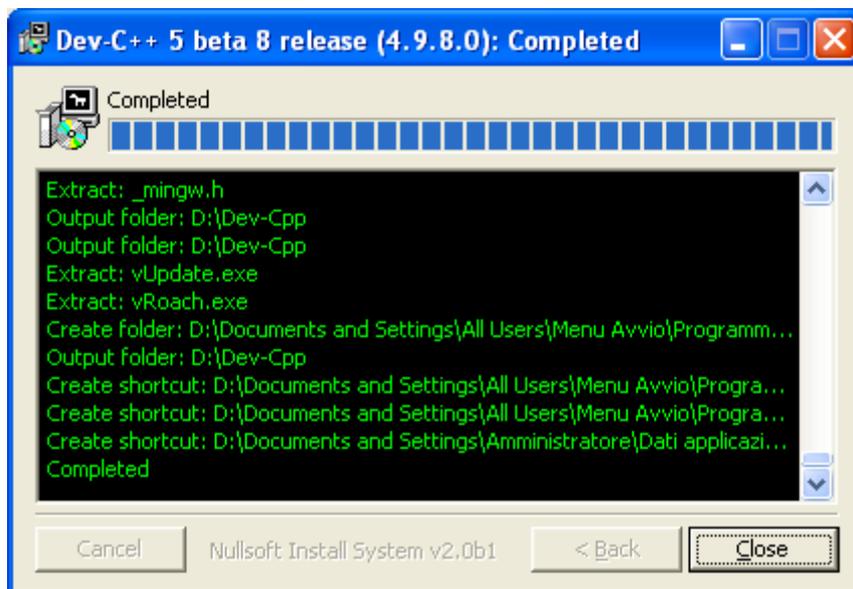


Fig. 2.4: Estrazione terminata

Si installa poi l'aggiornamento alla versione 4.9.8.10.

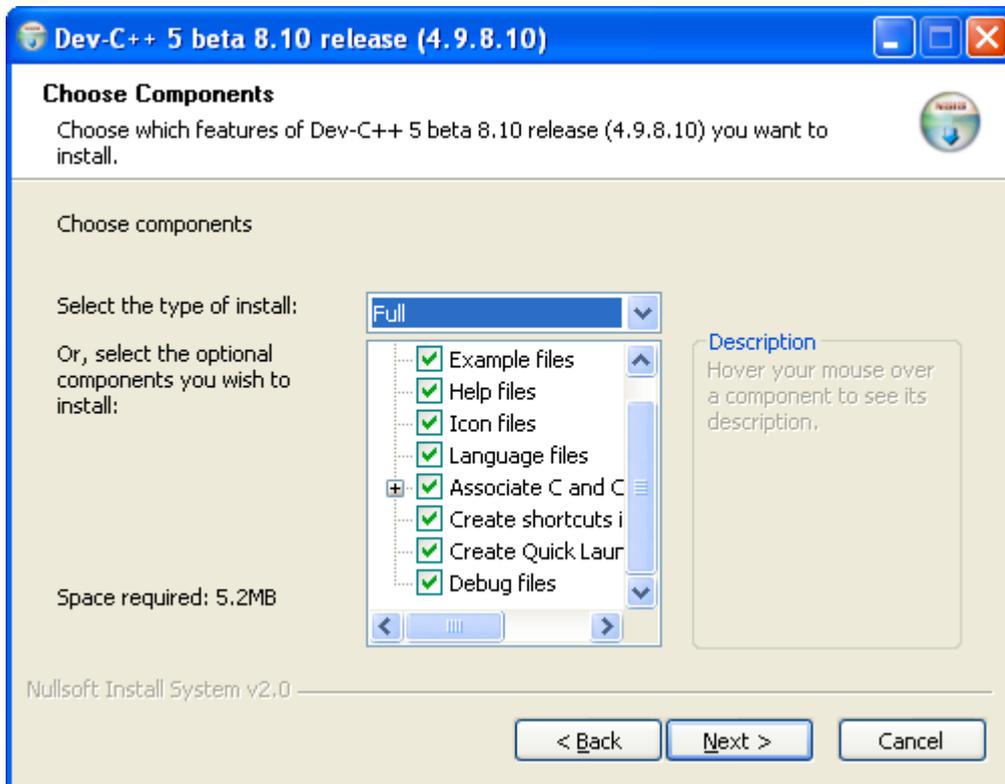


Fig. 2.5: Installazione aggiornamento 4.9.8.10

## 2.2. Configurazione

La configurazione iniziale di DevC++ consente di scegliere la lingua e l'aspetto grafico.

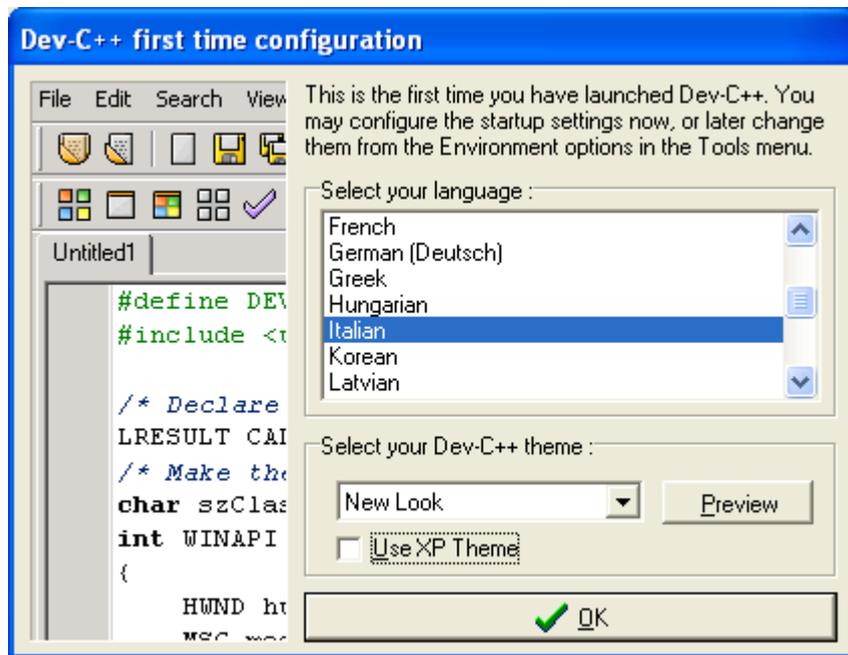


Fig. 2.6: Configurazione iniziale

È inoltre possibile configurare i parametri di compilazione e personalizzare l'editor del codice sorgente.

### 2.2.1. Opzioni di compilazione

Ci interessa la compilazione di programmi C e non C++.

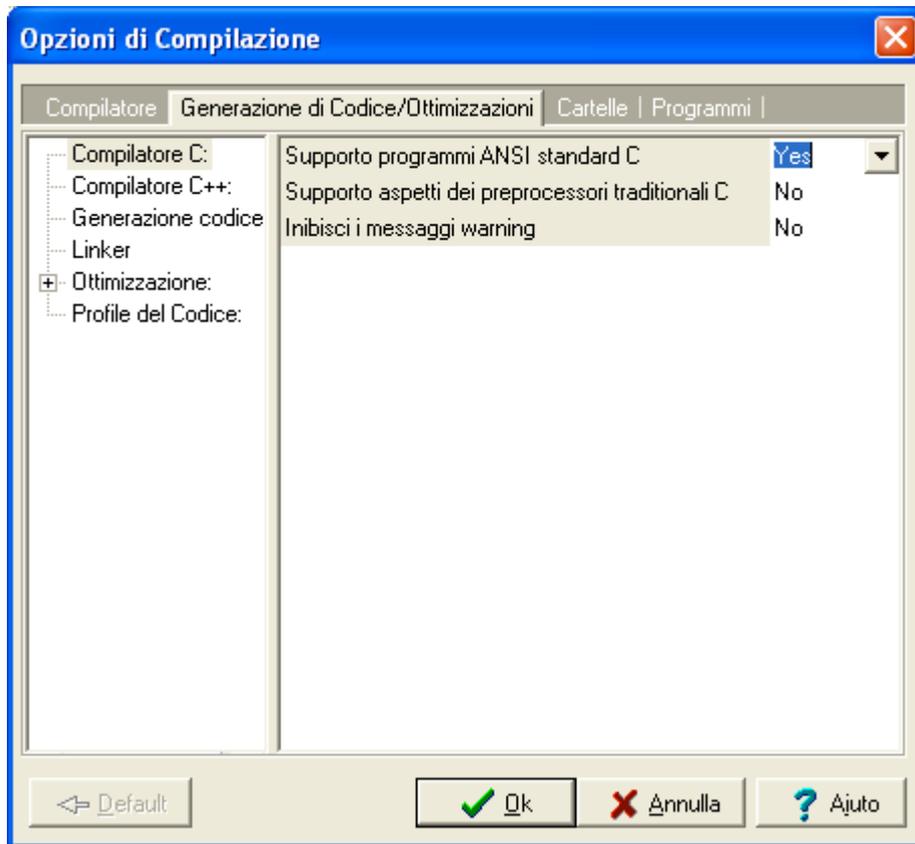


Fig. 2.7: Opzioni di compilazione

## 2.2.2. Opzioni dell'editor

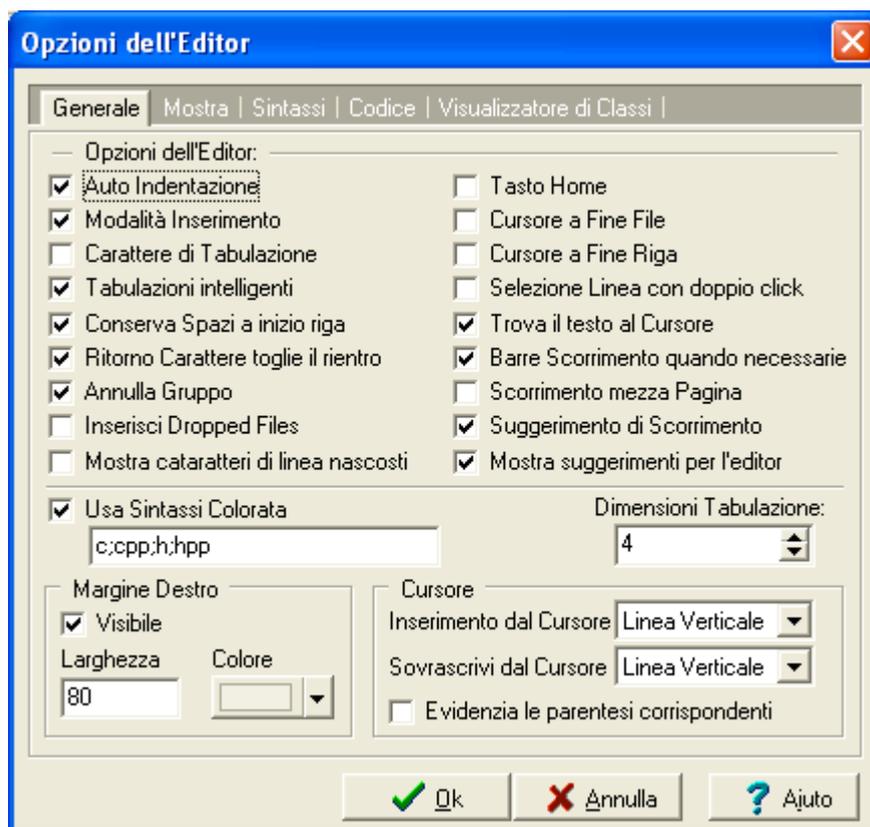


Fig. 2.8 : Opzioni dell'editor

### 3. Sessione di lavoro

Una tipica sessione di lavoro con un ambiente di sviluppo prevede una serie di azioni, spesso svolte in modo ciclico.

Durante una sessione di lavoro:

1. si crea un nuovo progetto o si apre un progetto esistente;
2. si modifica il codice sorgente;
3. si salvano le modifiche apportate;
4. si effettua la compilazione;
5. in caso di esito positivo, si effettua il link (se necessario); altrimenti si ritorna al passo 2;
6. in caso di esito positivo il programma viene eseguito, altrimenti si ritorna al passo 2.

#### 3.1. Il progetto

Dal momento che un programma può essere costituito da più file, l'IDE prevede il concetto di progetto, ossia un gruppo di file di codice sorgente in linguaggio C o C++ (.C) che costituisce il programma da compilare ed eseguire.

La creazione di un file progetto è indispensabile per la creazione di file di codice sorgente.

Se il programma risiede in un unico file (ad esempio *esercizio.c*), è necessario creare dapprima il progetto che lo contiene.

##### 3.1.1. Creazione di un nuovo progetto

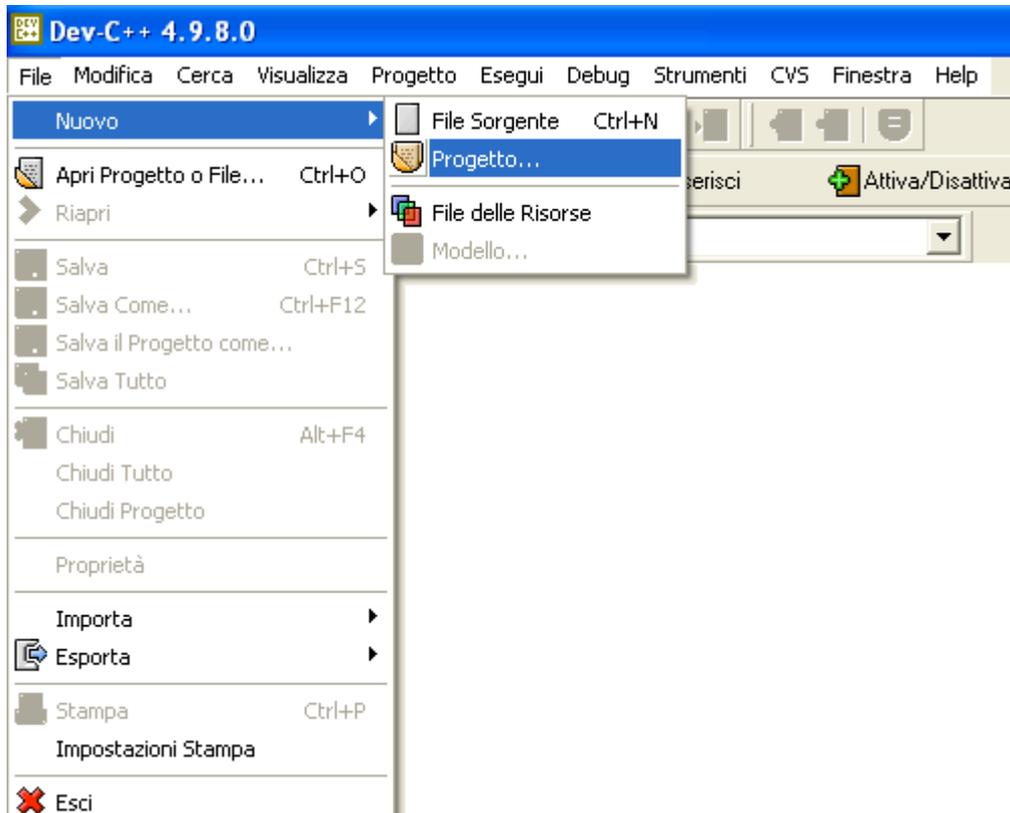


Fig. 3.1: Creazione di un nuovo progetto

Ci occuperemo di applicazioni console, cioè che utilizzano come standard input e standard output una finestra MS-DOS.

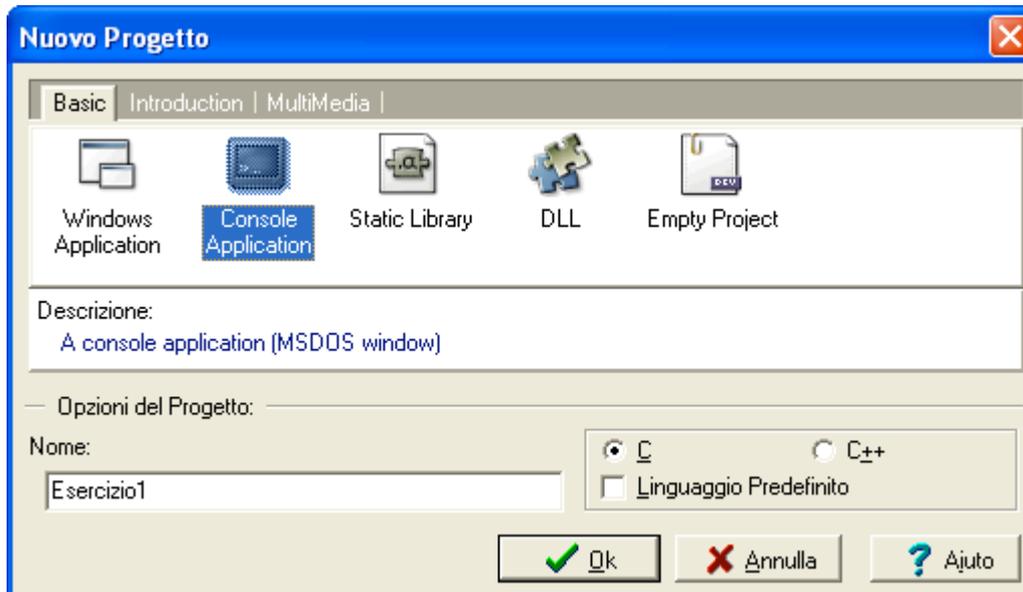


Fig. 3.2: Applicazioni console in C

Nel caso di progetti costituiti da un unico file, è possibile assegnare al progetto (file con estensione .dev) lo stesso nome del file sorgente (file con estensione .c).

### 3.1.2. Apertura di un progetto/file esistente

L'apertura di un file o progetto esistente consente di modificare il codice sorgente.

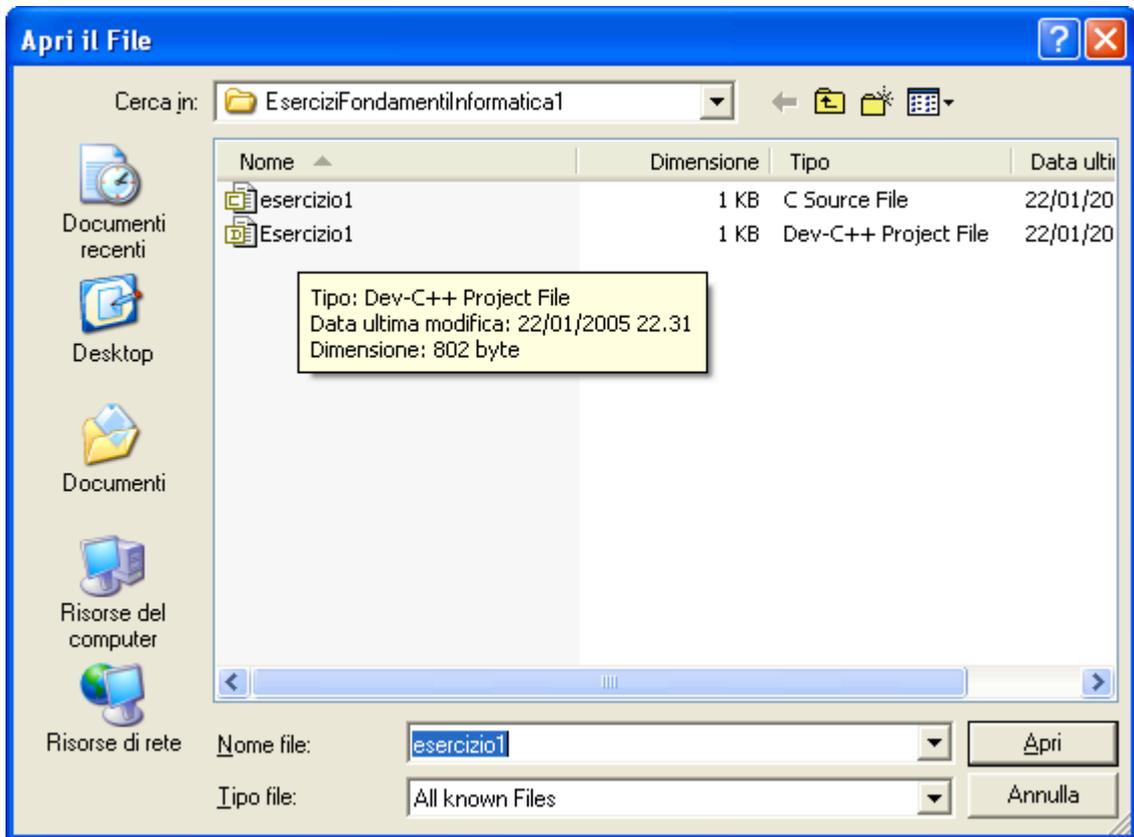


Fig. 3.3: Apertura di un progetto/file esistente

### 3.1.3. Modifica del codice sorgente

L'IDE dispone di un editor che, oltre ad offrire l'evidenziazione automatica delle parole chiave (keyword), permette di:

- effettuare ricerche e sostituzioni di testo;
- di spostarsi ad una determinata funzione del programma;
- di spostarsi ad una determinata riga (linea) del programma.

Queste funzionalità di ricerca/sostituzione testo sono accessibili dal menu *Cerca*.

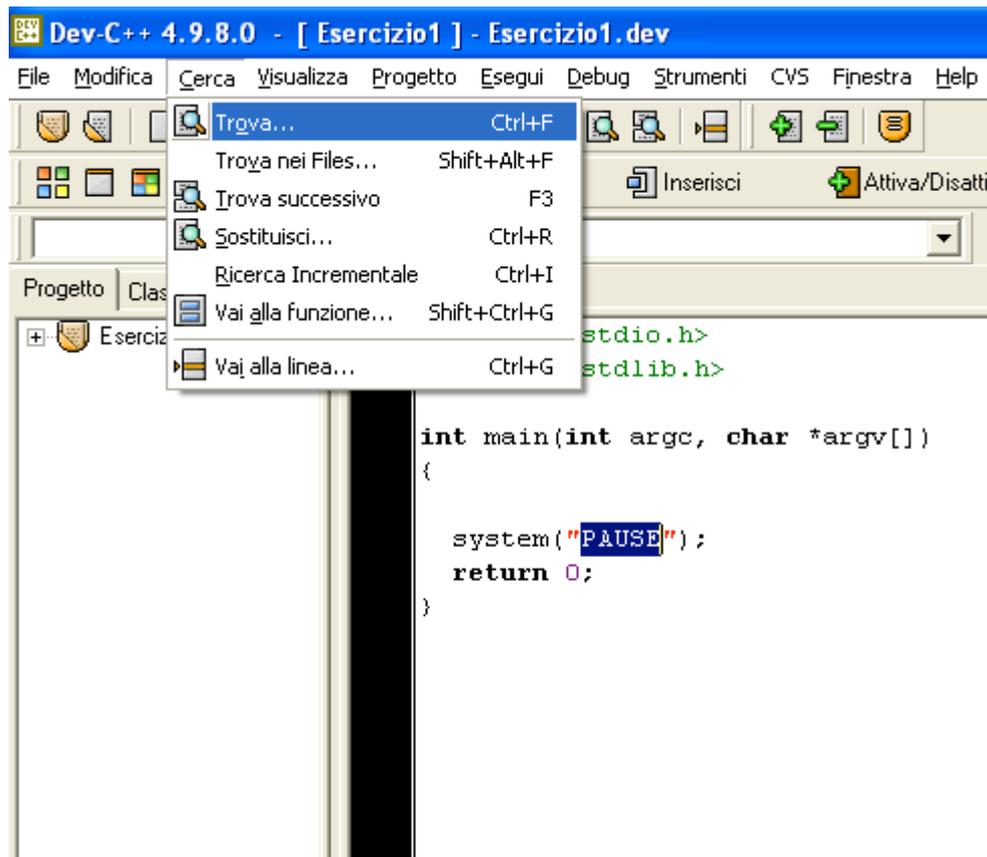


Fig. 3.4: Ricerca di testo

I risultati della ricerca in più files del progetto si trovano nella apposita scheda (tab) della finestra di output.

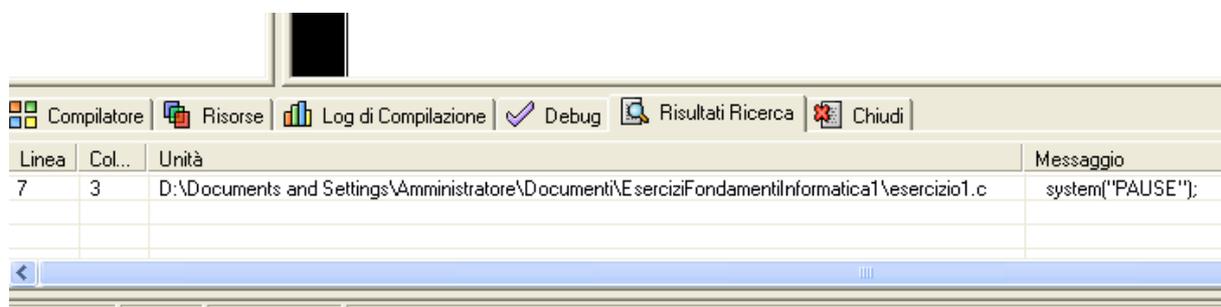


Fig. 3.5: Risultati ricerca su più file

## 3.2. Esecuzione del programma

### 3.2.1. Compilazione

La voce *Compila* consente di effettuare la compilazione ed il link del progetto, ossia dell'insieme di file di codice sorgente .C di cui è costituito.

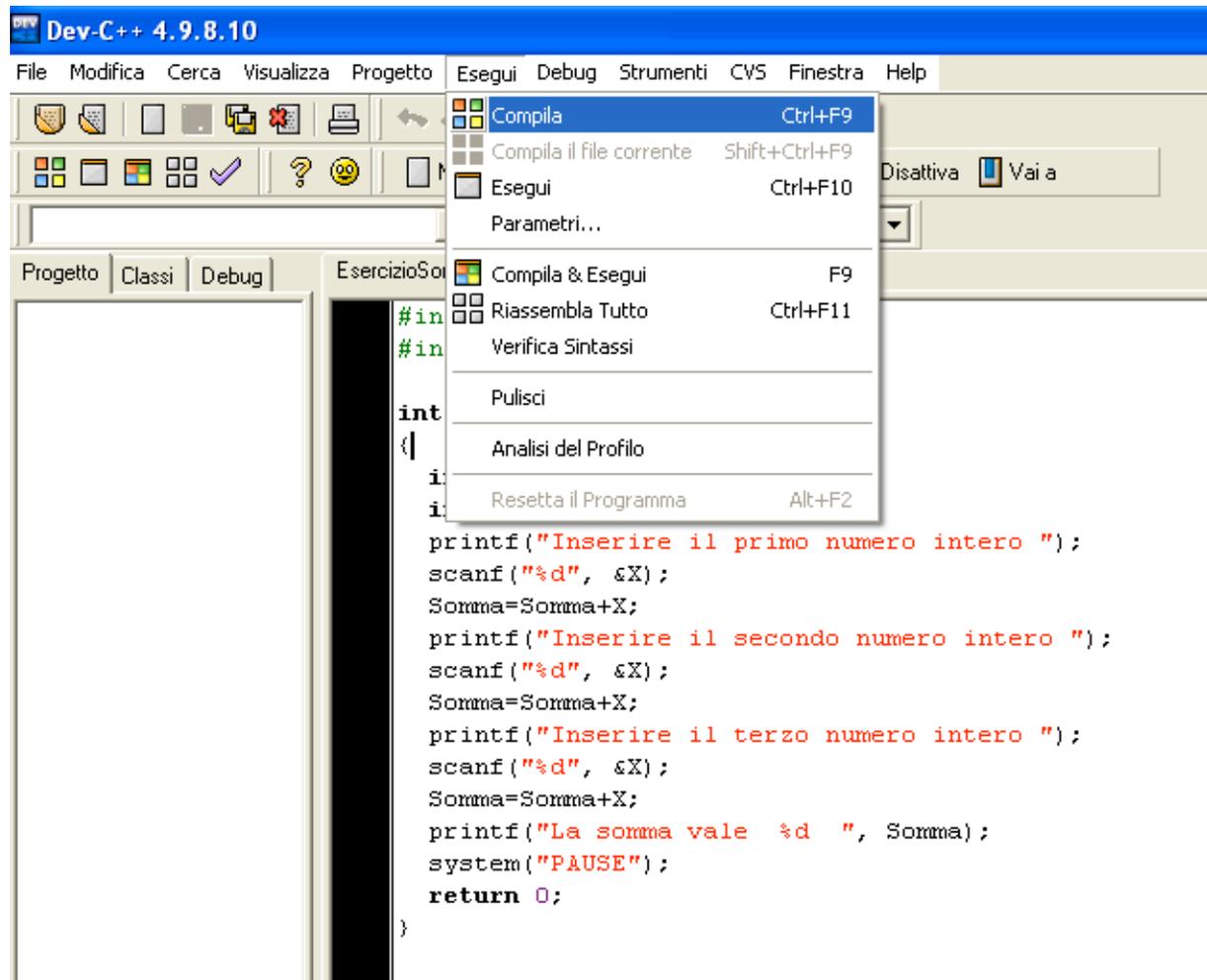


Fig. 3.6: *Compilazione*

Il comando comporta il salvataggio automatico del progetto.

Se la compilazione è avvenuta con successo compare la finestra di dialogo *Compile Progress* con status "Done."

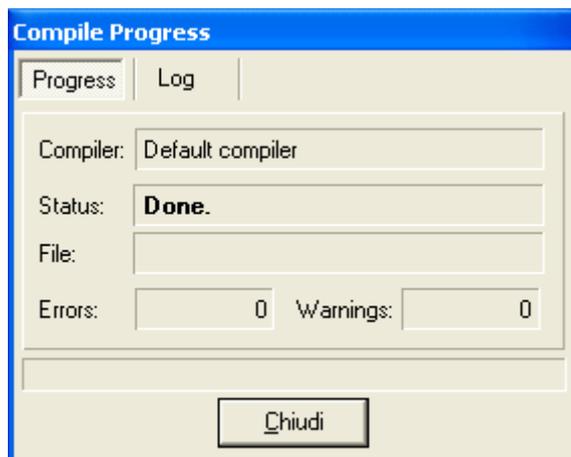


Fig. 3.7: Progetto compilato

Gli eventuali errori di compilazione sono presentati nella scheda (tab) *Compilatore*.

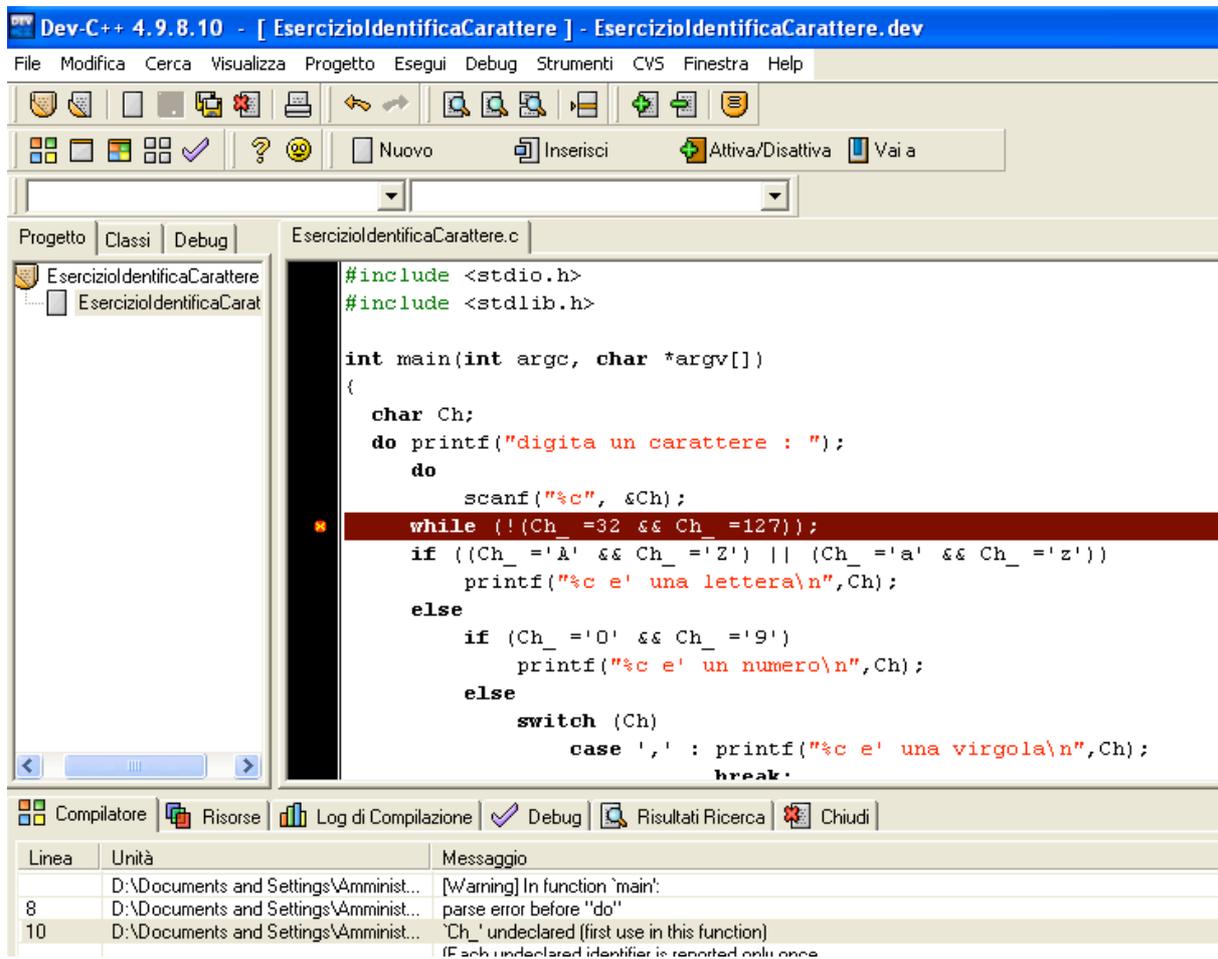


Fig. 3.8: Scheda "Compilatore" con errori di compilazione

Nella scheda (tab) *Log di compilazione* si possono vedere, indipendentemente dalla presenza di errori o meno, i risultati (log) dell'attività di compilazione.



Fig. 3.9: Scheda "Log di compilazione"

### 3.2.2. Compilazione, link ed esecuzione

L'esecuzione del comando *Compila & Esegui* comporta nell'ordine:

1. compilazione del file sorgente *main.c* (generazione del file di codice oggetto *main.o*);
2. qualora siano presenti, compilazione dei rimanenti file sorgenti *.c* componenti il progetto (generazione dei rispettivi file di codice oggetto *.o*);
3. link del progetto (generazione del file *progetto.exe*).

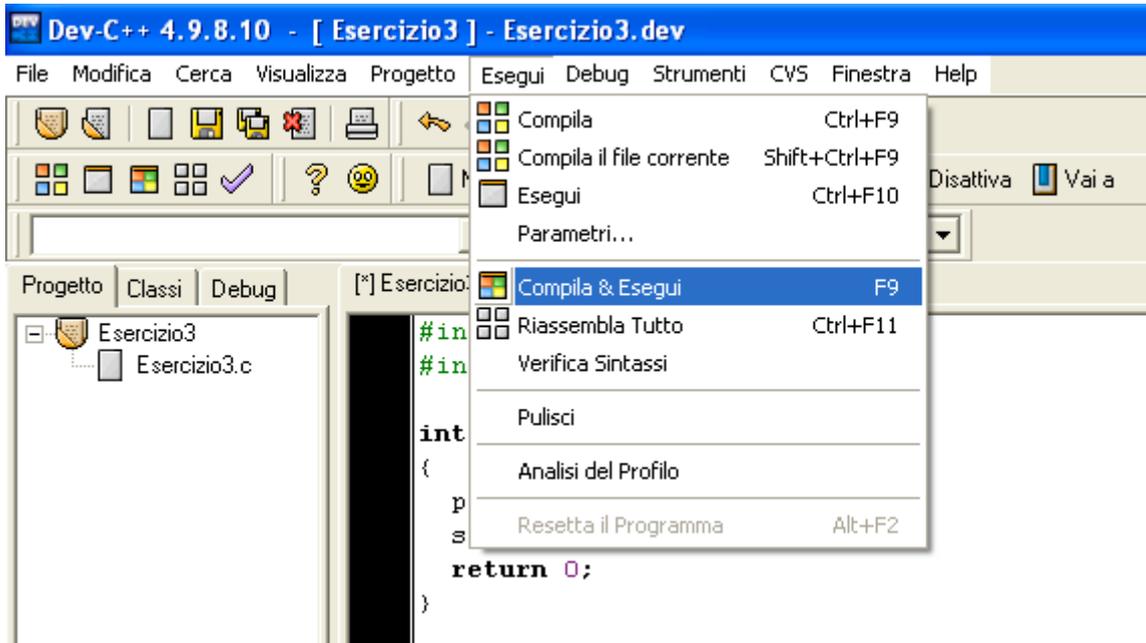


Fig. 3.10: Compilazione, link ed esecuzione

Il comando comporta il salvataggio automatico del progetto.

In presenza di errori il programma non viene eseguito.

In presenza di errori, nella scheda (tab) *Compilazione* si possono vedere le tipologie di errore riscontrate (compilazione, link e build).

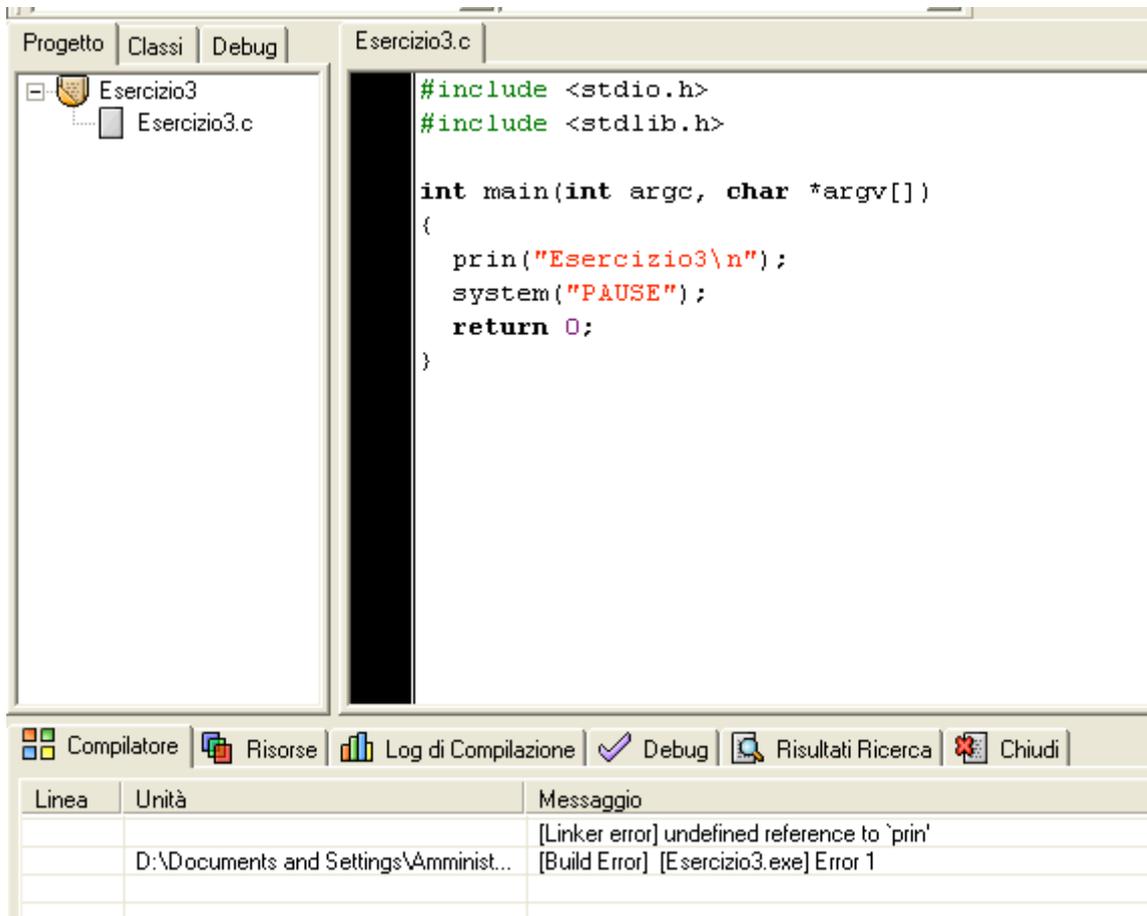


Fig. 3.11: Scheda "Compilatore" con errori di link e build

Nella scheda (tab) *Log di compilazione* si possono vedere i risultati (log) delle attività di compilazione e di link. In essa sono riportati gli eventuali errori generati.

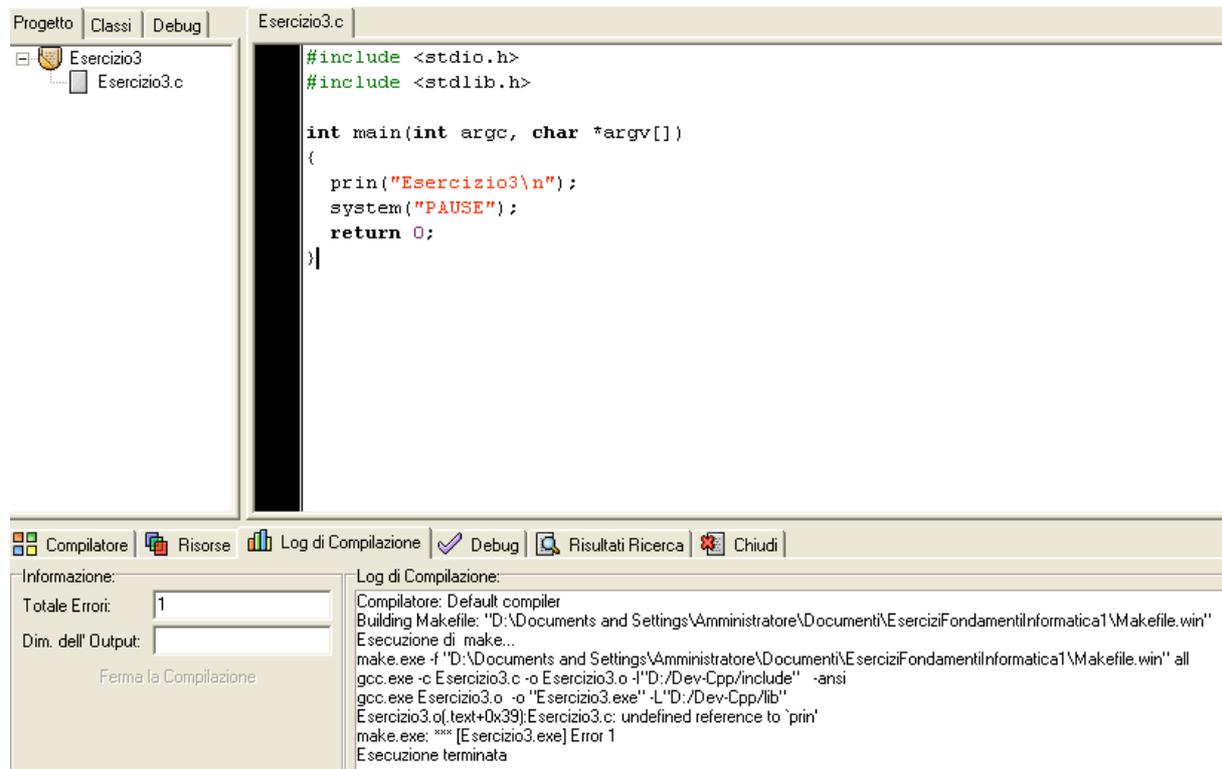


Fig. 3.12: Scheda "Log di compilazione" con errori

Con il comando *Pulisci* si elimina dalla directory corrente il file .exe generato. Si consiglia di utilizzare questo comando prima di una nuova azione di compilazione ed esecuzione.

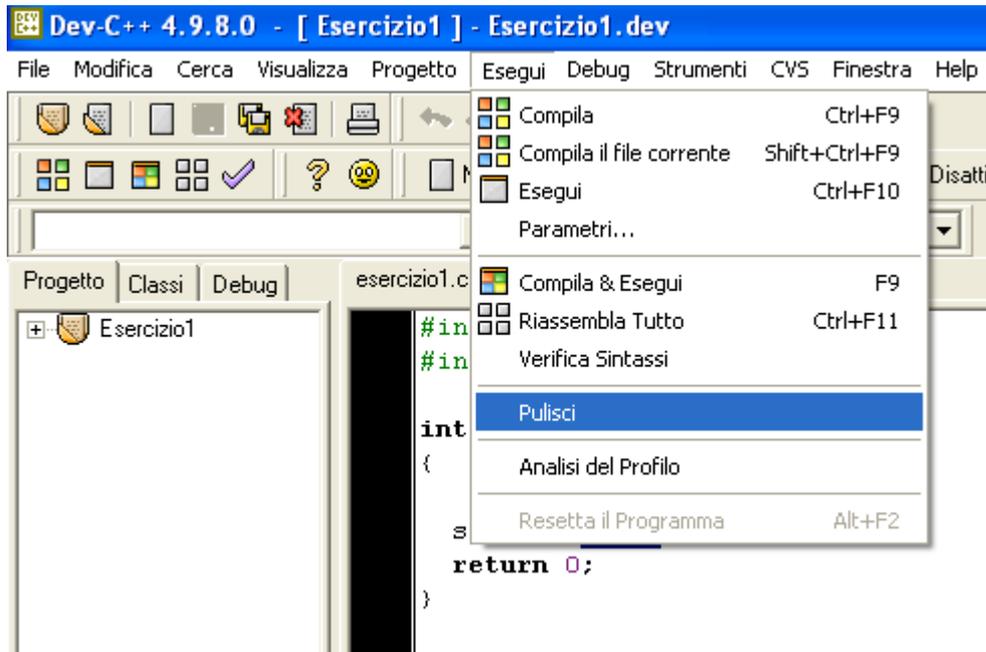


Fig. 3.13: Eliminazione dell'eseguibile generato

### 3.2.3. Esecuzione con parametri

È possibile parametrizzare l'esecuzione di un programma attribuendo un valore ai parametri della funzione main.

```
int main (int argc, char *argv[])
```

Per specificare il valore di tali parametri si deve richiamare il comando *Parametri...* dal menu *Esegui*. I parametri (stringa) devono essere separati da spazio.

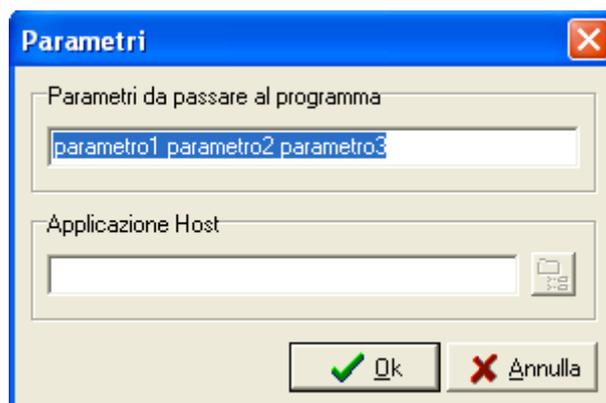


Fig. 3.14: Specifica di parametri nella funzione main()

### 3.3. Salvataggio del codice

I comandi di salvataggio sono contenuti nel menu *File*.

Le opzioni di salvataggio sono 4:

- salvataggio del file sorgente .C corrente (comando *Salva*);
- salvataggio con rinomina del file sorgente .C corrente (comando *Salva Come*);
- salvataggio con rinomina del progetto (.dev) corrente (comando *Salva il Progetto come...*);
- salvataggio di tutti i file aperti (*Salva Tutto*).

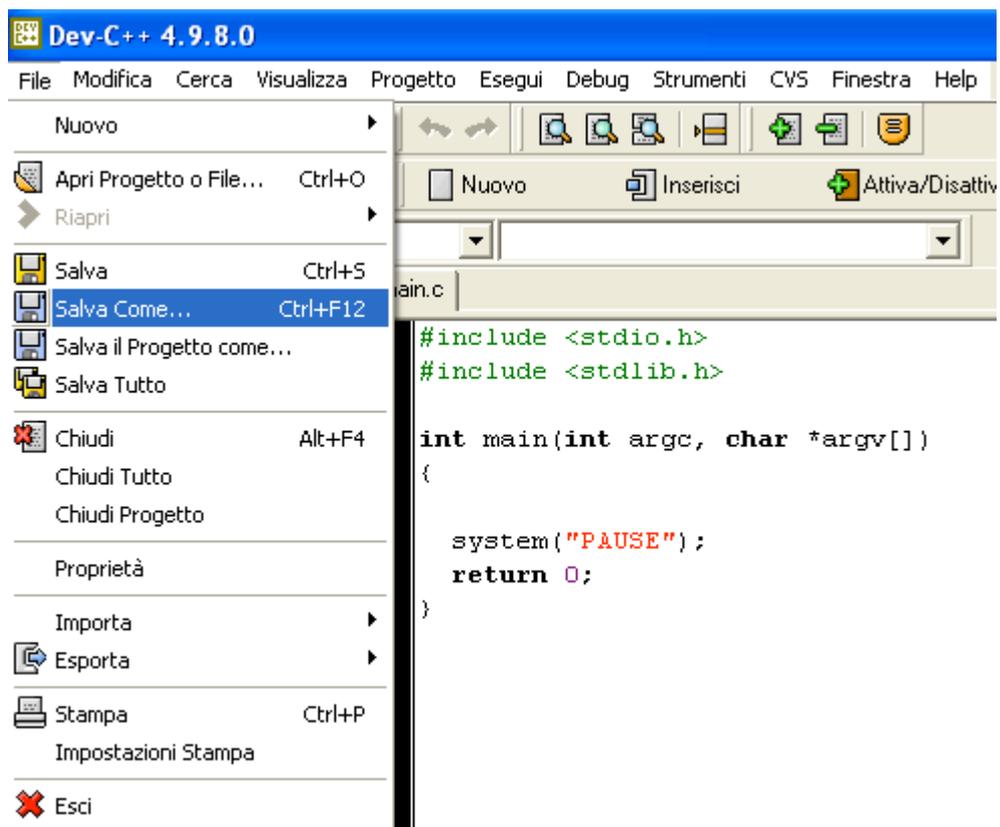


Fig. 3.15: Salvataggio

Il salvataggio con nome prevede di specificare la directory in cui il file deve essere salvato.

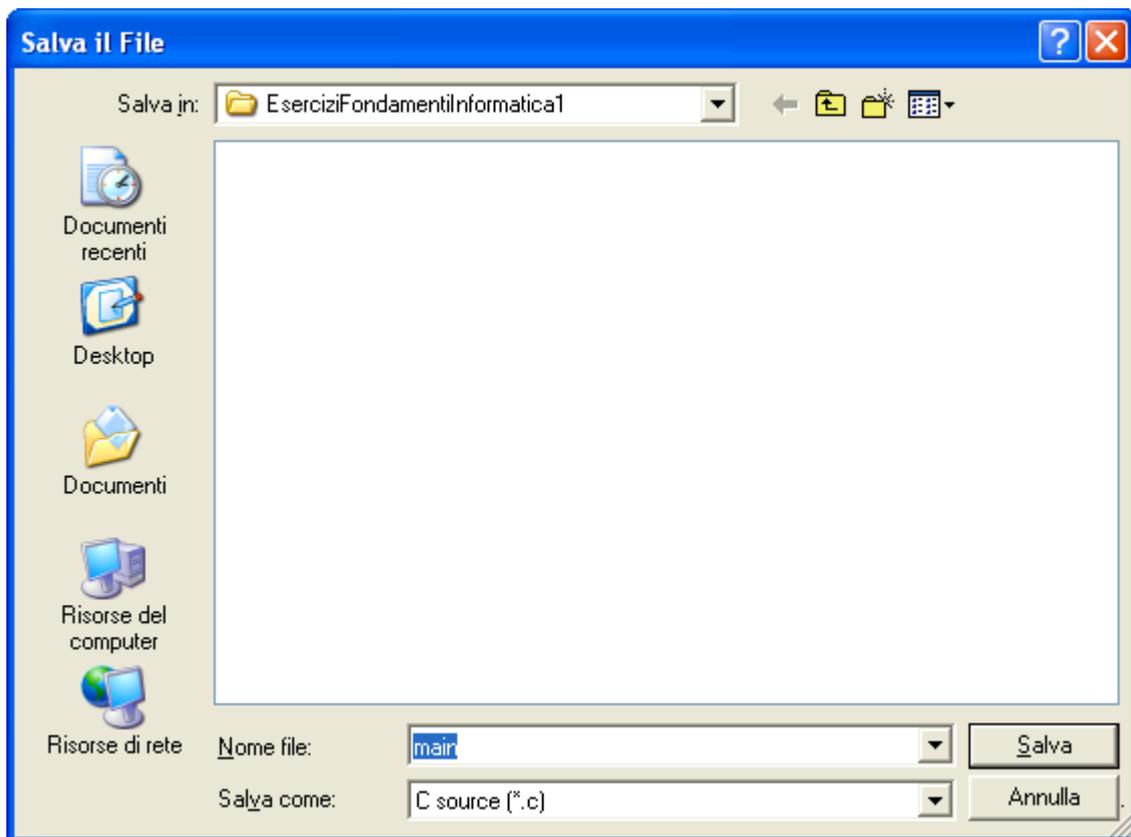


Fig. 3.16: Salvataggio con nome di un file sorgente .C

## 4. Alcuni riferimenti utili

### 4.1. Help in linea

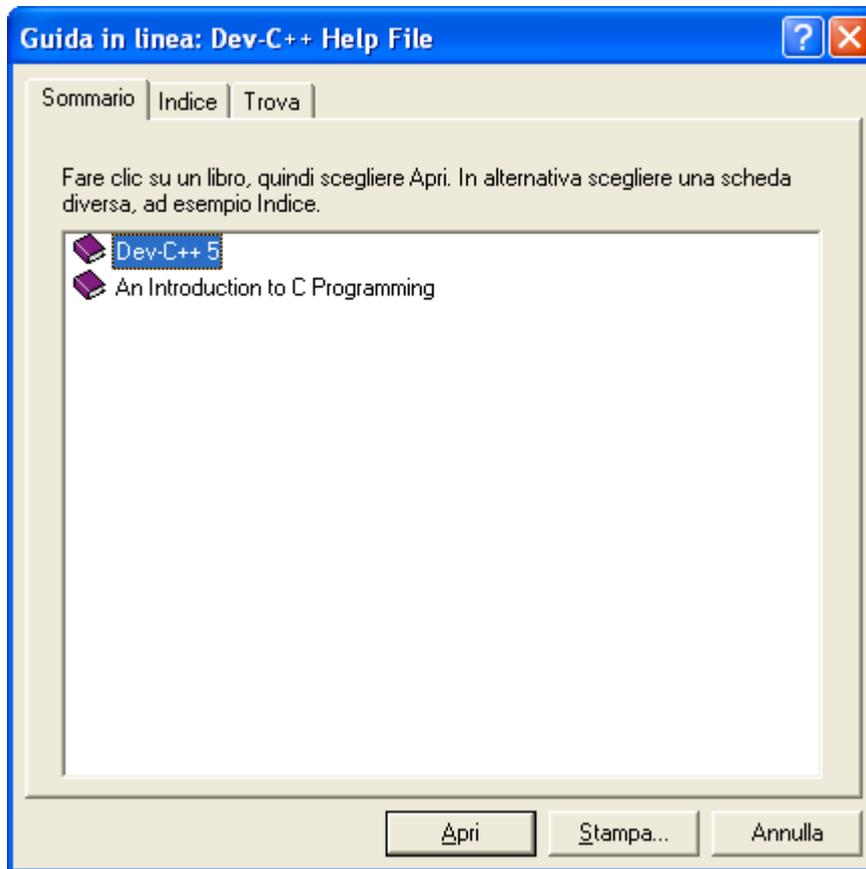


Fig. 4.1: *Help in linea*

### 4.2. Bloodshed.net

È il sito ufficiale di presentazione del compilatore DevC++ in lingua inglese.

<http://www.bloodshed.net/dev/devcpp.html>

### 4.3. SourceForge.NET

È un sito di sviluppo in inglese che tratta anche del compilatore DevC++.

Dalla pagina <http://sourceforge.net/projects/dev-cpp/> è possibile:

- scaricare il compilatore;
- scaricare aggiornamenti (patches & bug fixes);
- scaricare informazioni sulle precedenti versioni del compilatore (release notes);
- consultare forum;
- consultare l'elenco dei bugs;
- consultare news sullo stato della documentazione, sulle versioni, sulla soluzione di problemi, ecc..

#### **4.4. Forum di sviluppatori**

È uno forum di sviluppatori DevC++ in lingua italiana.

<http://forum.redangel.it/>