

1 – Passi preliminari

Avviare l'ambiente di sviluppo **Visual Studio 2015** utilizzando il collegamento presente sul desktop.

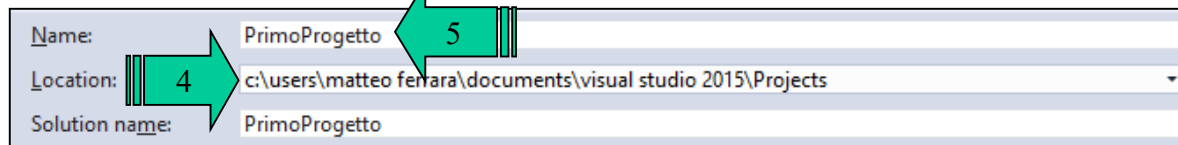
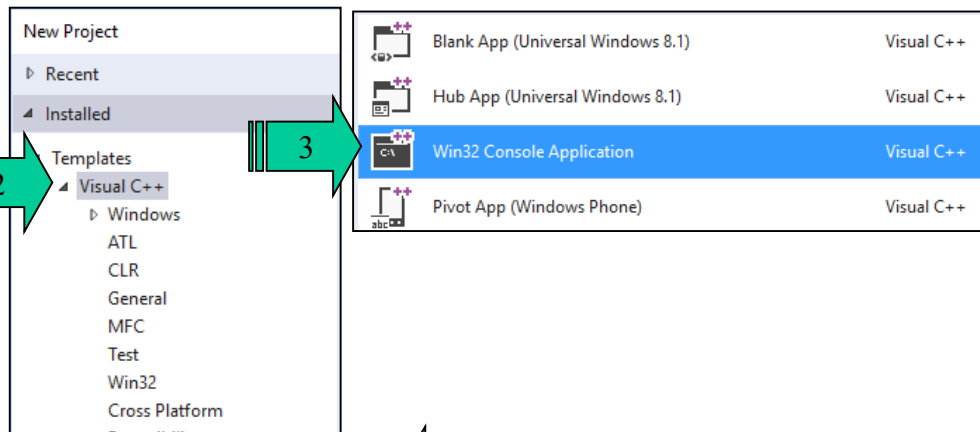
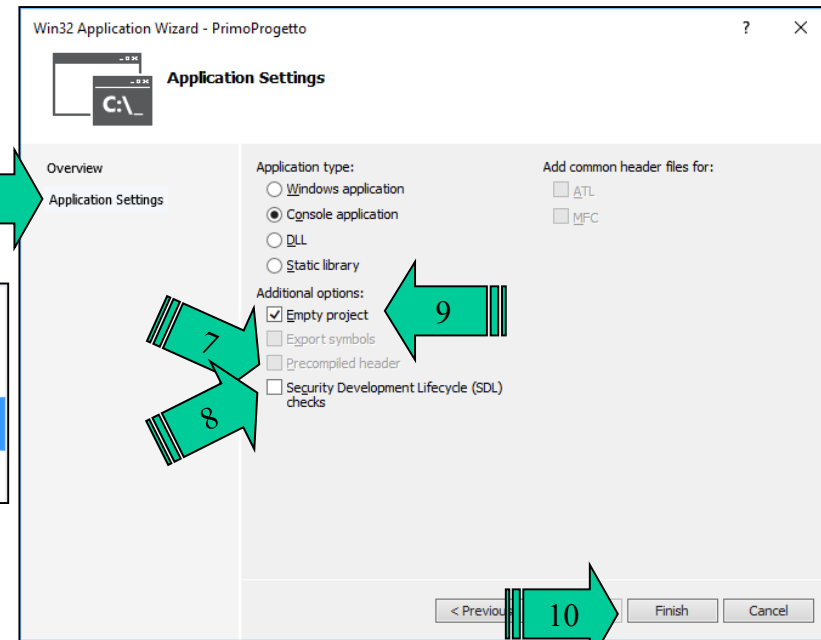
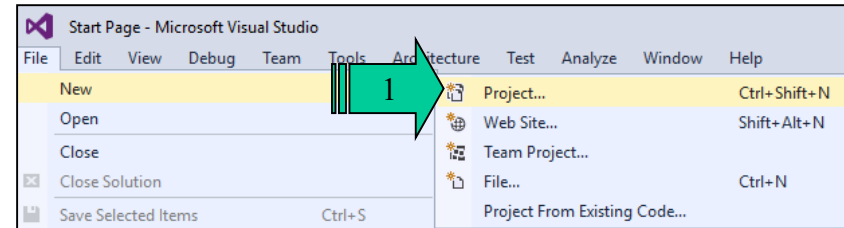
La prima volta che si avvia Visual Studio, si deve impostare la personalizzazione dell'ambiente di sviluppo: **Visual C++**.



ATTENZIONE: alla prima esecuzione Visual Studio potrebbe richiedere un po' di tempo prima di avviarsi.

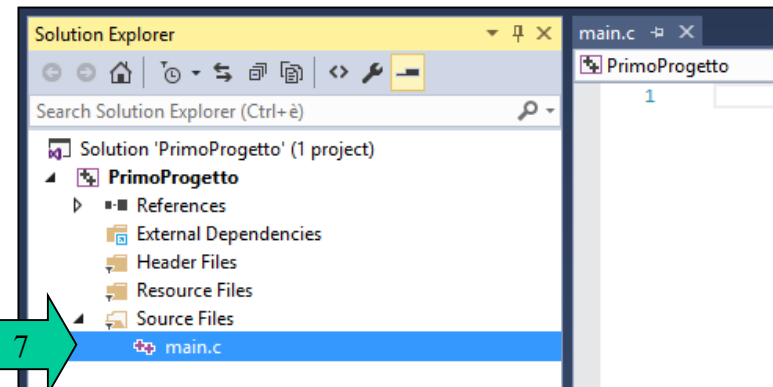
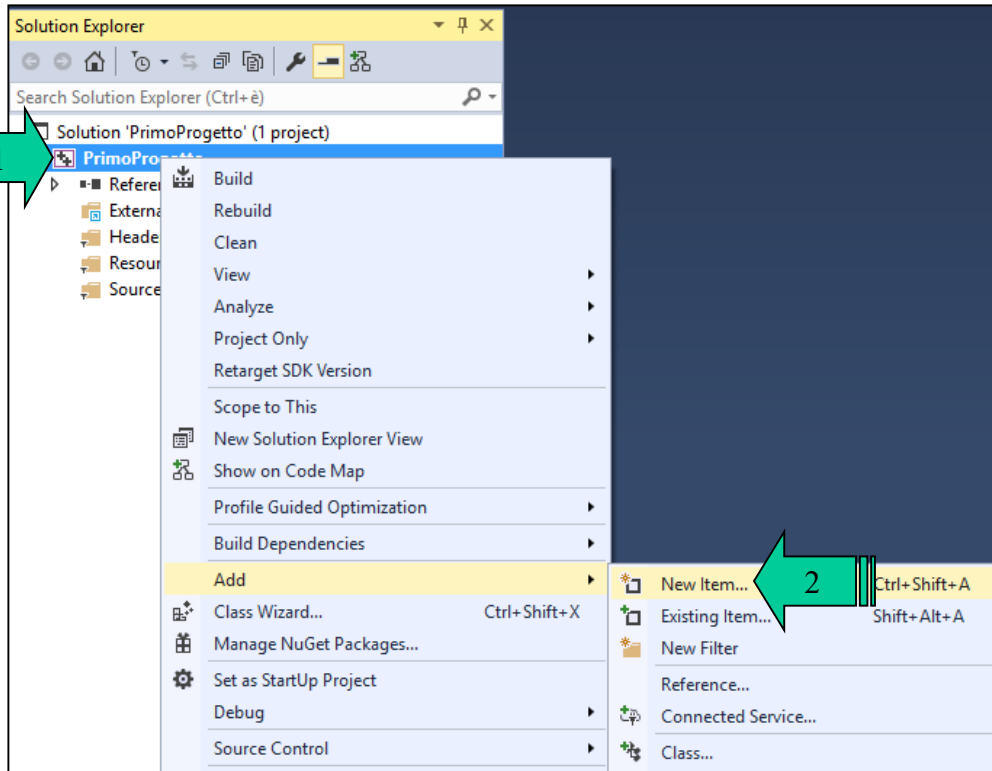
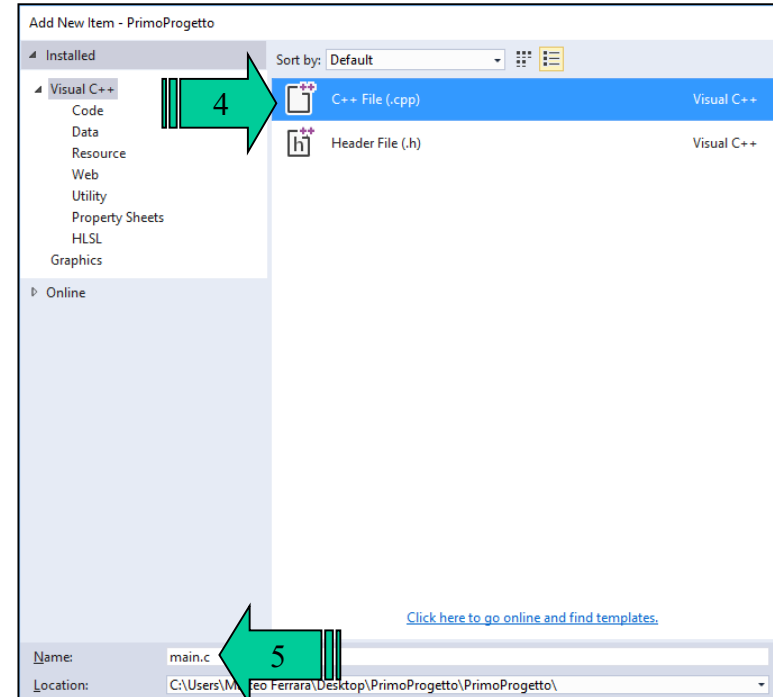
2 – Creazione di un nuovo progetto

1. Selezionare “New→Project” dal menù File
2. Selezionare la voce “Visual C++” dal menù di sinistra
3. Selezionare “Win32 Console Application” dal menù centrale
4. Verificare che la *location* del progetto sia sotto la cartella “Documents” del vostro utente
5. Digitare il nome del progetto e premere OK
6. Selezionare “Application Settings”
7. Deselezionare “Precompiled header”
8. Deselezionare “Security Development Lifecycle”
9. Selezionare “Empty project”
10. Premere “Finish”



3 – Inserire un file nel progetto

1. Fare click con il tasto destro sul progetto
2. Selezionare “Add → New Item” dal menù contestuale
3. Selezionare la voce “Visual C++” dal menù di sinistra
4. Selezionare “C++ File (.cpp)” dal menù centrale
5. Specificare il nome e l'estensione del file (Es. main.c)
6. Premere “Add”
7. Il file appena creato comparirà nella cartella “Source Files” del progetto



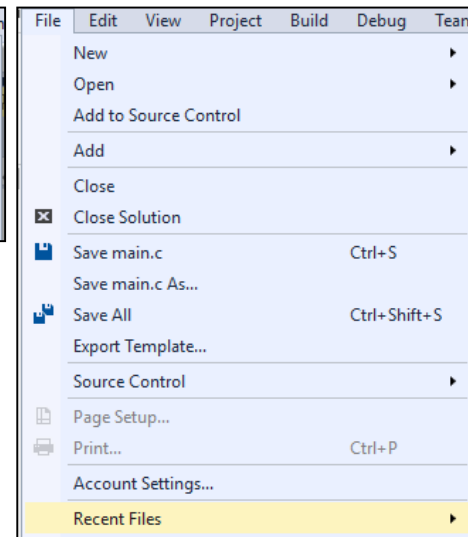
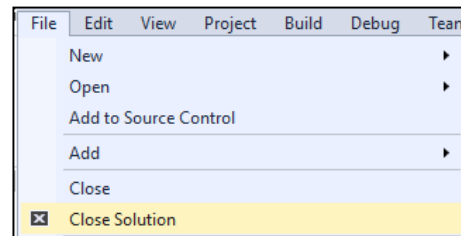
4 – Esercizi

•Directory e progetti

- Il Visual C++ crea automaticamente una cartella con lo stesso nome del progetto, in cui mette tutti i file
- Utilizzare il comando “Close Solution” del menù File per chiudere un progetto e aprirne un altro
- Creare un secondo progetto, seguendo gli stessi passi appena visti

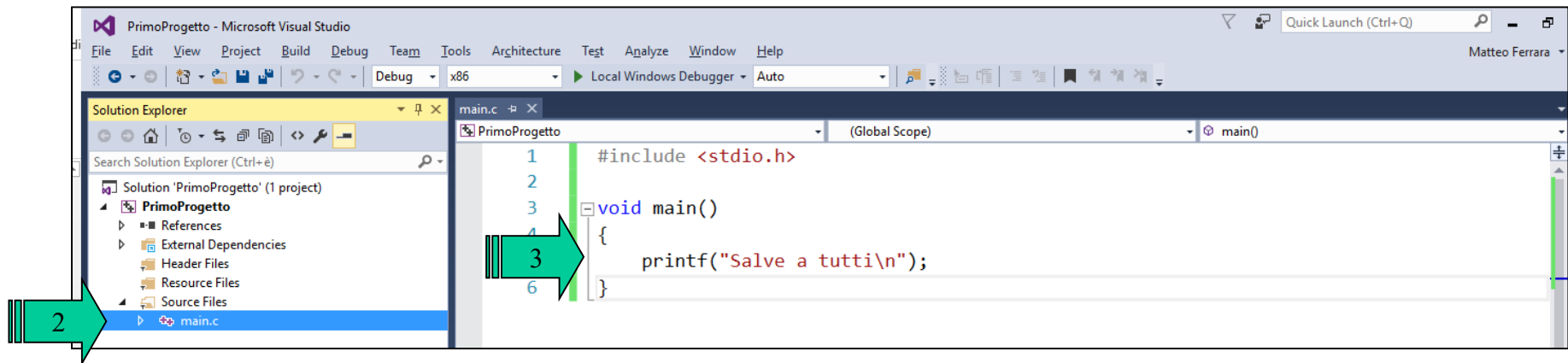
N.B. È importante mantenere fin dall’inizio un certo ordine nel creare programmi e progetti: ogni esercizio dovrà essere costituito da un diverso progetto situato in un’apposita cartella; il nome del progetto dovrebbe essere legato al testo (o al numero) dell’esercizio.

Suggerimento: In “Recent Projects” del menù “File” si trovano gli ultimi progetti aperti



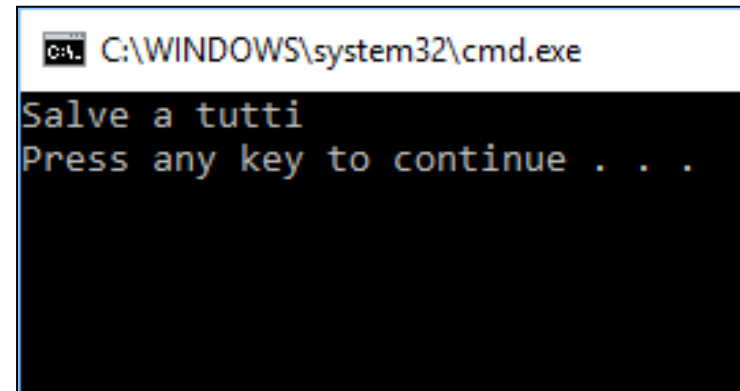
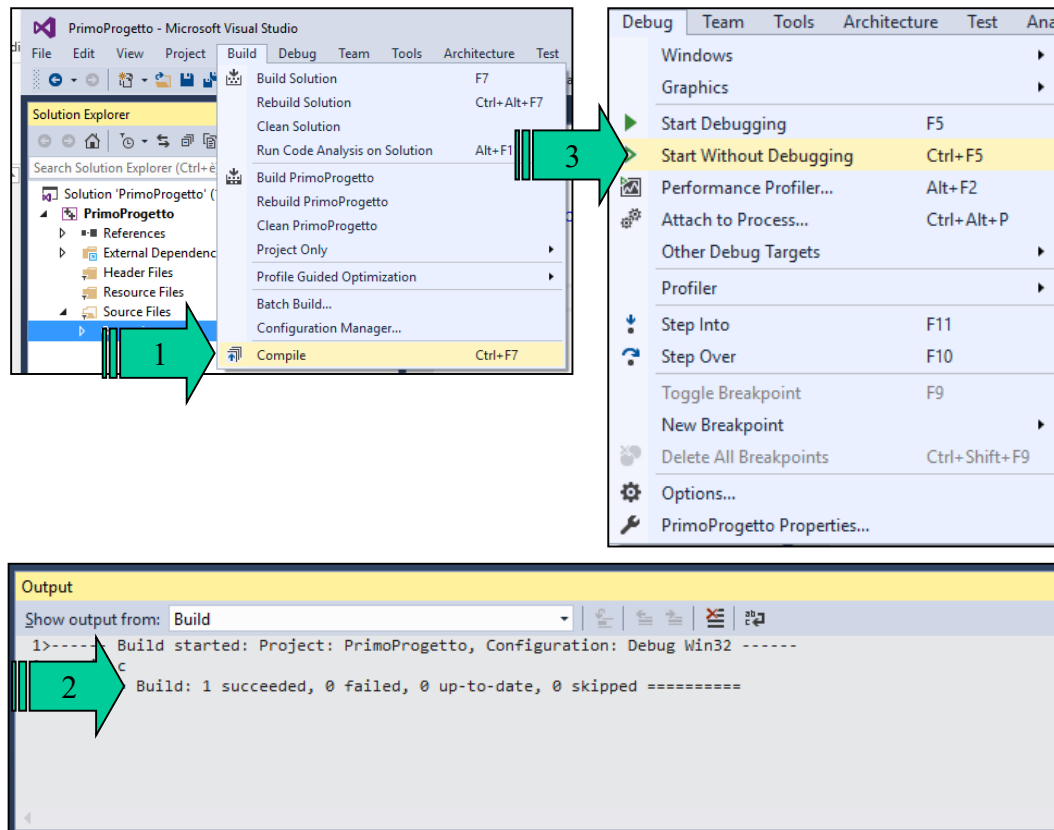
5 – Un primo programma in C

1. Aprire il progetto creato in precedenza
2. Doppio click sul file .c nell'albero del progetto
3. Scrivere il programma riportato in figura



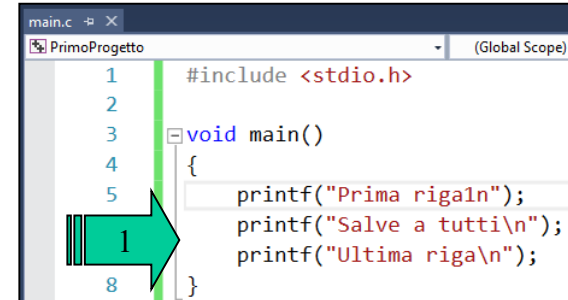
6 – Compilare ed eseguire il programma

1. Selezionare dal menù Build la voce “Compile” (Ctrl+F7) per compilare e linkare il programma
2. Controllare il risultato nella finestra di output (scheda “Build”)
3. Selezionare “Start Without Debugging” (Ctrl+F5) dal menù Debug per eseguire il programma

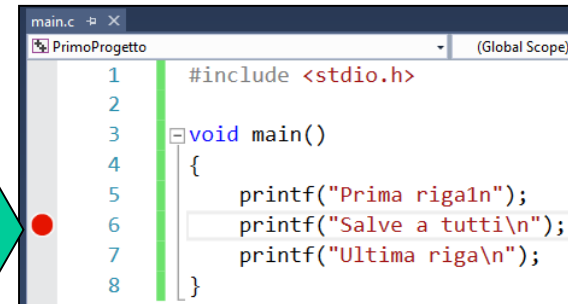


7 – Debug del programma

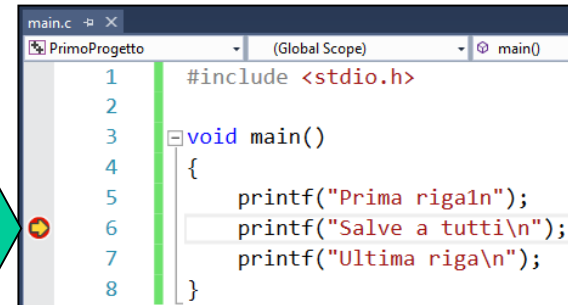
1. Modificare il programma come mostrato in figura
2. Cliccare sulla barra grigia a sinistra della seconda chiamata “printf” per inserire un breakpoint
3. Selezionare “Start Debugging” (F5) dal menù Debug per eseguire il programma: l’esecuzione si interromperà in corrispondenza del breakpoint
4. Selezionare “Step Over” (F10) dallo stesso menù per avanzare di una riga alla volta
5. Selezionare “Continue” (F5) per lasciare proseguire il programma fino alla fine



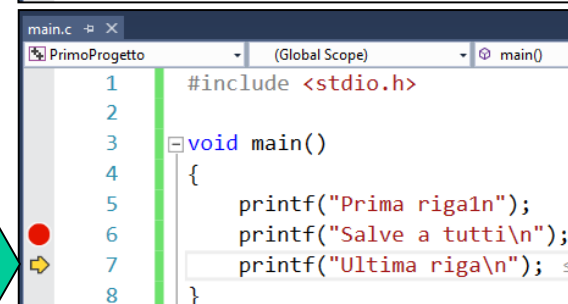
```
main.c - X
PrimoProgetto (Global Scope)
1  #include <stdio.h>
2
3  void main()
4  {
5      printf("Prima riga\n");
6      printf("Salve a tutti\n");
7      printf("Ultima riga\n");
8  }
```



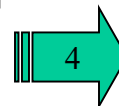
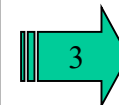
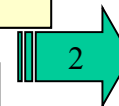
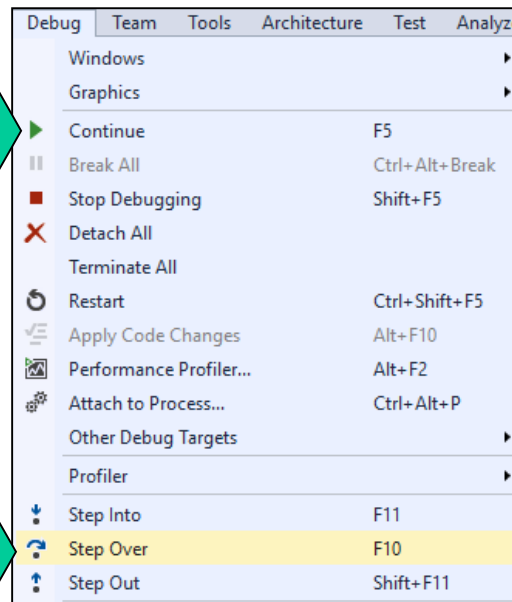
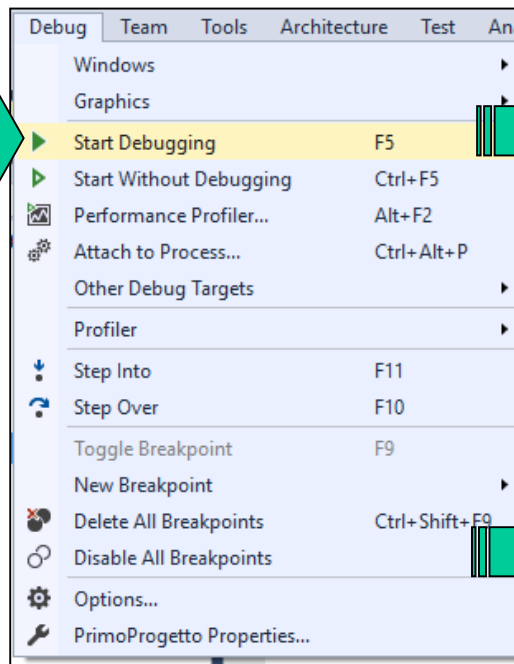
```
main.c - X
PrimoProgetto (Global Scope)
1  #include <stdio.h>
2
3  void main()
4  {
5      printf("Prima riga\n");
6      printf("Salve a tutti\n");
7      printf("Ultima riga\n");
8  }
```



```
main.c - X
PrimoProgetto (Global Scope) main()
1  #include <stdio.h>
2
3  void main()
4  {
5      printf("Prima riga\n");
6      printf("Salve a tutti\n");
7      printf("Ultima riga\n");
8  }
```



```
main.c - X
PrimoProgetto (Global Scope) main()
1  #include <stdio.h>
2
3  void main()
4  {
5      printf("Prima riga\n");
6      printf("Salve a tutti\n");
7      printf("Ultima riga\n");
8  }
```



8 – Esercizi

- **Compilazione**

- Modificare il contenuto delle stringhe stampate a video, compilare ed eseguire nuovamente il programma

- **Debugging**

- Fare pratica con i comandi di debug esaminati, in quanto saranno ampiamente utilizzati in seguito (N.B. Per rimuovere un breakpoint, cliccarci sopra con il mouse)

- **Ambiente di sviluppo**

- Il Visual C++ è un ambiente di programmazione molto ricco di funzionalità (la maggior parte delle quali non sarà utilizzata nelle esercitazioni): fare pratica esplorando i menù e le toolbar. Ad esempio provare a: aprire e chiudere file e progetti, modificare la posizione di toolbar e finestre, visualizzare nuove toolbar, stampare un file .c aperto, ...
- Fare pratica con le funzionalità di “copia” e “incolla” dell’editor del Visual C++ per ricopiare parti del codice senza doverle riscrivere (nello stesso file e fra file diversi)