Le librerie utente in C

Prof. Rio Chierego

23 marzo 2024

Librerie utente in C: introduzione

- In informatica, una libreria, o più raramente biblioteca, è un insieme di funzioni o strutture dati predefinite e predisposte per essere riutilizzate da altri programmi software attraverso un'opportuna procedura di collegamento.

- Il termine libreria nasce da una errata trasposizione lessicale dell'inglese library (letteralmente, biblioteca), ma ormai è così diffuso nel vocabolario dei professionisti informatici da essere accettato quale possibile traduzione.

- Lo scopo delle librerie software è fornire una collezione di entità di base pronte per l'uso ovvero riuso di codice, evitando al programmatore di dover riscrivere ogni volta le stesse funzioni o strutture dati e facilitando così le operazioni di sviluppo e manutenzione.

- Le librerie possono contenere sia codice eseguibile che oggetti a seconda dell'ambiente e del tipo di linguaggio di programmazione che si utilizza.

Librerie utente in C: vantaggi

I vantaggi principali derivanti dall'uso di una una `libreria` all'interno di un programma sono i seguenti:

- si può separare la logica di programmazione di una certa applicazione da quella necessaria per la risoluzione di problemi specifici, quali il calcolo di funzioni matematiche o la gestione delle collezioni;
- le entità definite in una certa libreria possono essere riutilizzate da più applicazioni (riuso);
- si può modificare la libreria separatamente dal programma, senza limiti alla potenziale vastità di funzioni e strutture dati man mano disponibili nel tempo.

Ogni linguaggio di programmazione possiede la sua collezione di `librerie`, le quali vengono distinte in librerie standardizzate e librerie non standardizzate. La differenza tra librerie standard e non standard influisce in maniera determinante sulla portabilità di un programma software fra sistemi operativi diversi e piattaforme hardware diverse.

Librerie utente in C: static library

Quando il collegamento (binding) viene eseguito durante la creazione di un eseguibile o di un altro file oggetto, è noto come collegamento statico o collegamento anticipato (early-binding).

In questo caso, il collegamento viene solitamente eseguito da un linker, ma può anche essere eseguito dal compilatore.

Una libreria statica (o static library) è pensata per essere collegata staticamente.

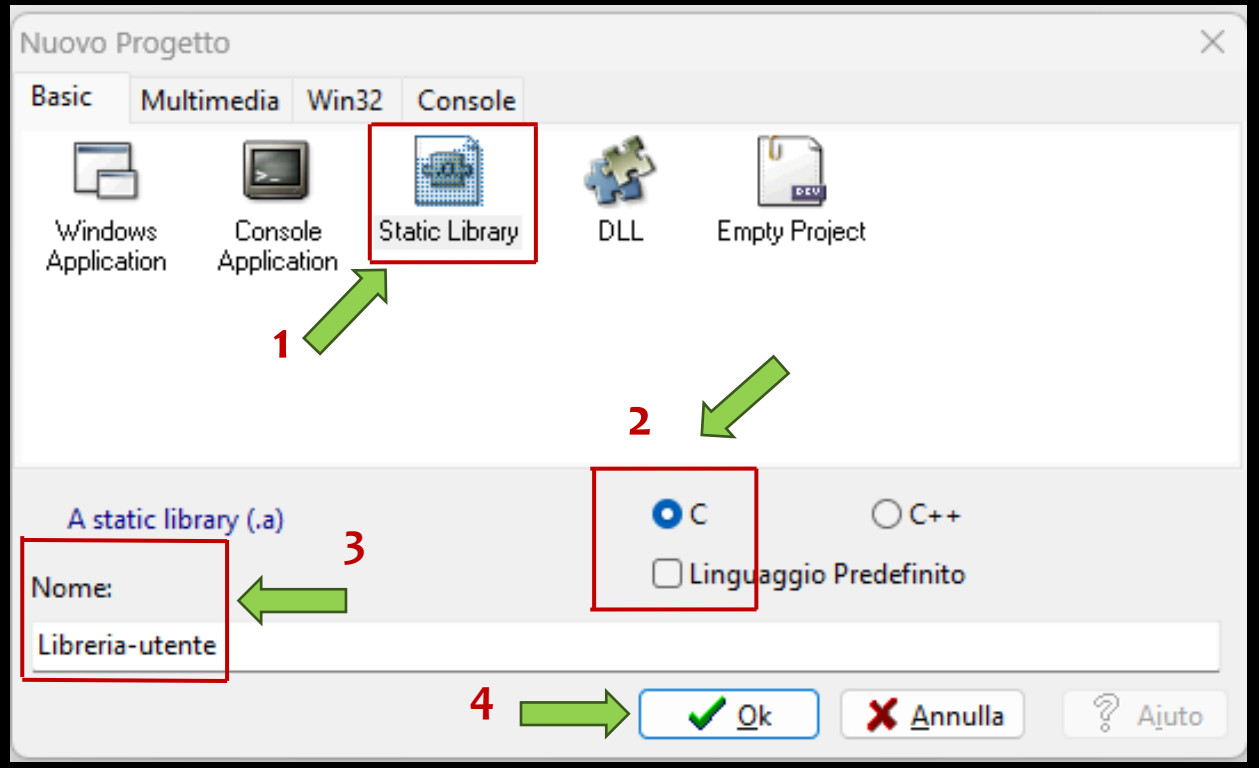
In origine esistevano solo librerie statiche. Il collegamento statico deve essere eseguito quando vengono ricompilati i moduli.

Tutti i moduli richiesti da un programma a volte sono collegati staticamente e copiati nel file eseguibile.

Questo processo e il file autonomo risultante è noto come costruzione statica del programma.

Come creare una libreria statica in DEV C++

- 1 Creare un Nuovo Progetto di tipo Static Library
- 2 Assicurarsi che sia selezionato, nel nostro caso, C (ma è la STESSA COSA anche con C++)
- 3 Chiamare il nuovo progetto Libreria-Utente
- 4 Salvarlo nella cartella desiderata



Come creare una libreria statica in DEV C++

- 5 Creare nel progetto Libreria-Utente il file SommaP.c contenente la procedura SommaP ()

```
/* Definizione procedura SommaP*/
void SommaP (int x, int y, int* somma)
{

*somma = x + y;

return;
}
```

- 6 Creare nel progetto Libreria-Utente il file SommaF.c contenente la funzione SommaF ()

```
/* Definizione funzione SommaF */
int SommaF (int x, int y)
{
int somma;

somma = x + y;

return somma;
}
```

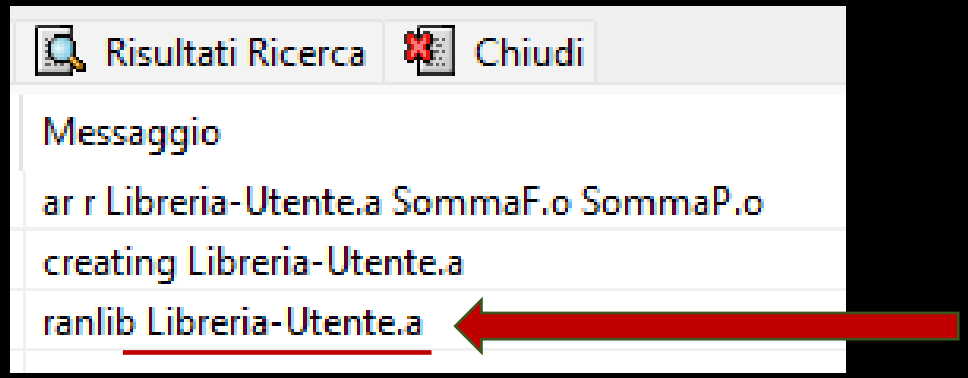
Come creare una libreria statica in DEV C++

- 7 Aggiungere al progetto Libreria-Utente il file prototipi.h contenente le segnature sia della procedura, sia della funzione

```
/* Dichiarazione prototipi prcedura SommaP e funzione SommaF */
void SommaP (int x, int y, int* somma);
int SommaF (int x, int y);
```

N.B. E' POSSIBILE NON INSERIRE IL FILE .H ALL'INTERNO DEL PROGETTO DELLA LIBRERIA STATICA, LASCIANDO IL COMPITO AL PROGETTO CHE LA UTILIZZERA'

- 8 Eseguire l'azione "Compila" sul progetto Libreria-Utente creando la libreria Libreria-Utente.a



Ora la nostra libreria statica è pronta… non ci resta che usarla!

Come usare una libreria statica in DEV C++

- 1 Creare un Nuovo Progetto di tipo Console Application
- 2 Assicurarsi che sia selezionato LO STESSO LINGUAGGIO scelto per la libreria (nel nostro caso C)
- 3 Dare un nome al nuovo progetto (nel nostro caso Uso-Libreria-Utente)
- 4 Salvarlo nella cartella desiderata

  N.B. nel nostro caso per semplicità NELLA MEDESIMA DIRECTORY dove è posizionato il progetto Libreria-Utente

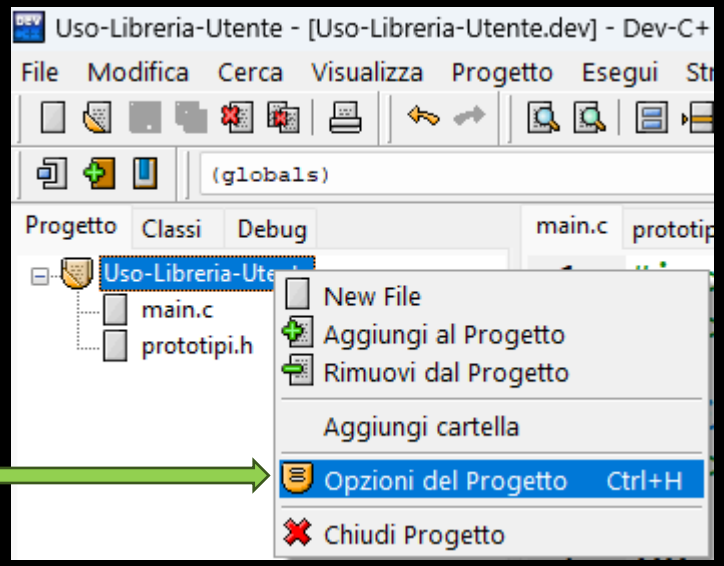
- 5 Aggiungere al progetto Uso-Libreria-Utente il file prototipi.h (SOLO se non è stato precedentemente incluso nel progetto della libreria statica)
- 6 Utilizzare nel file main.c del progetto le chiamate alla procedura SommaP ed alla funzion e SommaF

Come usare una libreria statica in DEV C++

- 7 Aggiungere nella sezione Linker del tab Parametri contenuto nelle Opzioni del Progetto il file creato in precedenza contenente la nostra Libreria-Statica.a

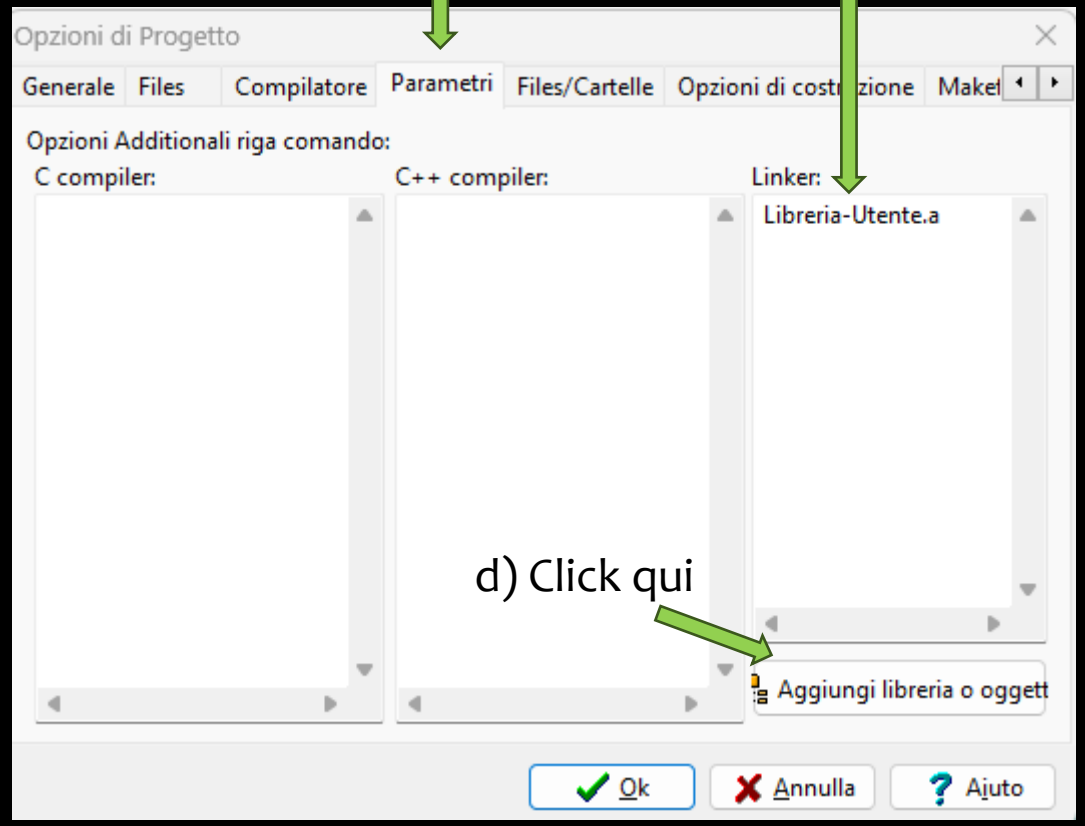
a) Tasto dx

b) Click qui



c) Scelta Tab

e) Aggiunta OK



d) Click qui

Come usare una libreria statica in DEV C++

- 8 Effettua l'azione "Compila" sul progetto Uso-Libreria-Utente avente come main.c un listato che invochi sia la procedura sia la funzione presenti nella libreria Libreria-Utente.a (come nel nostro esempio)

N.B. Se il file è stato già incluso nella libreria non va scritto

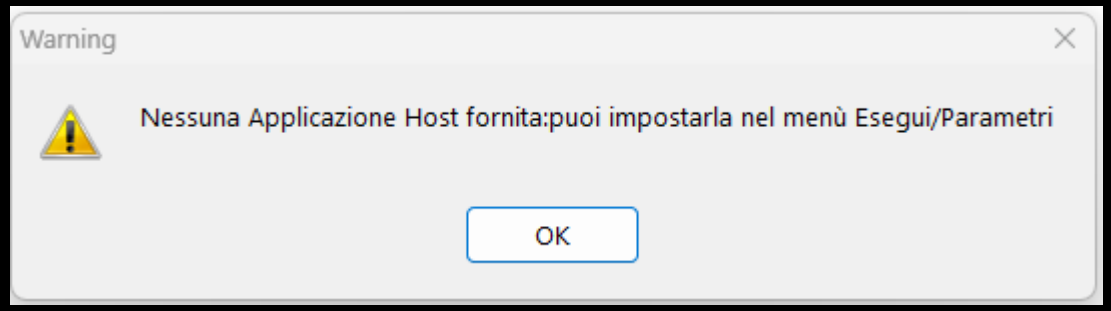
```c
#include <stdio.h>
#include <stdlib.h>

/* include file prototipi funzioni libreria utente */
#include "prototipi.h"

int main(int argc, char *argv[])
{
/* Dichiarazione variabili di input */
int a, b;
/* Dichiarazione variabili di output */
int s;

printf ("Inserire il valore di a: ");
scanf("%d", &a);
printf ("Inserire il valore di b: ");
scanf("%d", &b);

/* chiamata a procedura SommaP */
SommaP (a, b, &s);
printf("La somma calcolata con la PROCEDURA SommaP e': %d \n", s);

/* chiamata a funzione SommaF */
s = SommaF (a, b);
printf("La somma calcolata con la FUNZIONE SommaF e': %d \n", s);

return 0;
}
```

Chiamata alla PROCEDURA SommaP presente nella nostra libreria

Chiamata alla FUNZIONE SommaF presente nella nostra libreria

Come usare una libreria statica in DEV C++

N.B. Non effettuare mai né l'azione "Esegui" né l'azione "Compila & Esegui" in quanto OVVIAMENTE mancando di main si riceverà questo warning



Librerie utente in C: dynamic library

Una libreria dinamica (o dynamic library) è un archivio di codice eseguibile che può essere caricato da più processi contemporaneamente.

Questo genere di librerie viene collegato ai programmi in fase di esecuzione (late-binding) tramite funzioni dedicate del sistema operativo, in particolare il dynamic linker.

L'algoritmo di collegamento in fase di esecuzione è alquanto sofisticato e permette, tra le altre cose, di sostituire al volo il codice della libreria senza modificare l'applicazione che lo richiede.

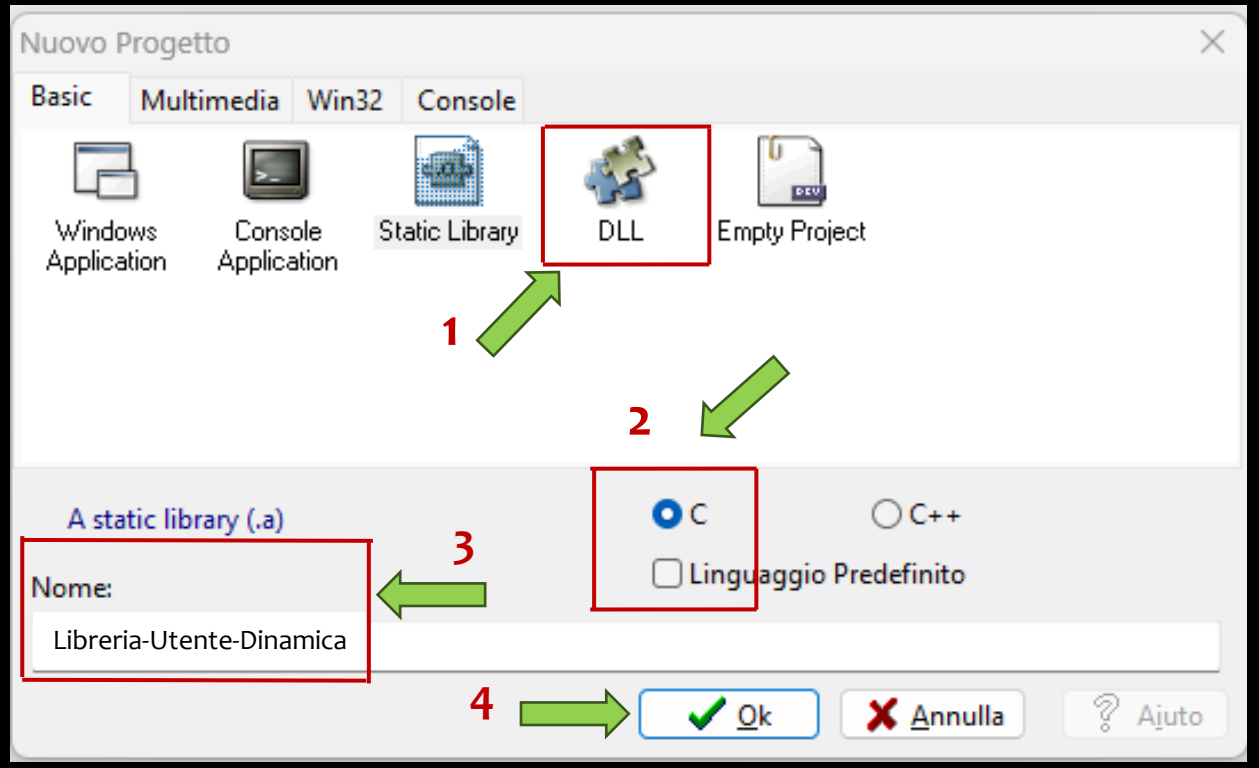
Per questo motivo si parla di collegamento dinamico (o late-binding) e per estensione di librerie dinamiche.

Esistono vari formati di librerie condivise tra cui i più famosi sono:

- DLL (Dynamic Link Library) utilizzato in Microsoft Windows
- ELF (Executable and Linkable Format) utilizzato in Linux, Unix e derivati.

Come creare una libreria dinamica in DEV C++

- 1 Creare un Nuovo Progetto di tipo DLL
- 2 Assicurarsi che sia selezionato, nel nostro caso, C (ma è la STESSA COSA anche con C++)
- 3 Chiamare il nuovo progetto Libreria-Utente-Dinamica
- 4 Salvarlo nella cartella desiderata



Come creare una libreria dinamica in DEV C++

- 5 Creare nel progetto Libreria-Utente-Dinamica il file SommaP.c contenente la procedura SommaP ()

```
/* Definizione procedura SommaP*/
void SommaP (int x, int y, int* somma)
{

*somma = x + y;

return;
}
```

- 6 Creare nel progetto Libreria-Utente-Dinamica il file SommaF.c contenente la funzione SommaF ()

```
/* Definizione funzione SommaF */
int SommaF (int x, int y)
{
int somma;

somma = x + y;

return somma;
}
```

Come creare una libreria dinamica in DEV C++

- 7 All'interno del progetto Libreria-Utente-Dinamica troveremo già il file dllmain.c così costituito

N.B. Se il file viene incluso qui non dovrà essere fatto nel progetto che utilizzerà la nostra libreria

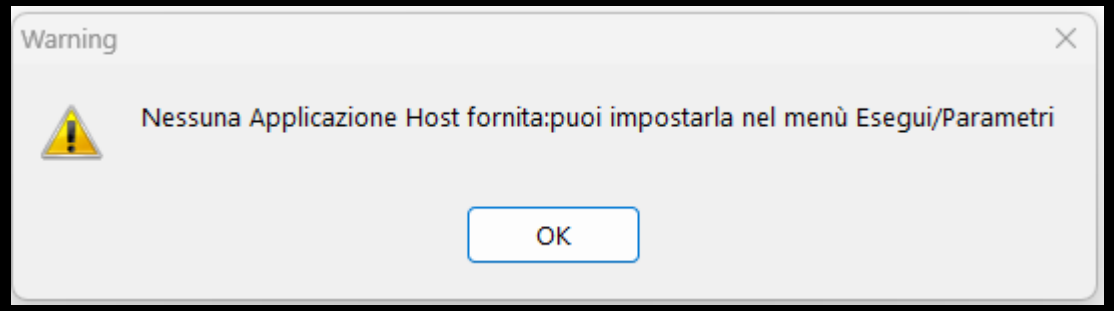
```
/* Replace "prototipi.h" with the name of your header */
//#include "prototipi.h"

#include <windows.h>

BOOL WINAPI DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
{
    switch(fdwReason)
    {
        case DLL_PROCESS_ATTACH:
        {
            break;
        }
        case DLL_PROCESS_DETACH:
        {
            break;
        }
        case DLL_THREAD_ATTACH:
        {
            break;
        }
        case DLL_THREAD_DETACH:
        {
            break;
        }
    }

    /* Return TRUE on success, FALSE on failure */
    return TRUE;
}
```

N.B. E' POSSIBILE NON INSERIRE IL FILE .H ALL'INTERNO DEL PROGETTO DELLA LIBRERIA DINAMICA, LASCIANDO IL COMPITO AL PROGETTO CHE LA UTILIZZERA'

Come creare una libreria dinamica in DEV C++

- 8 Eseguiamo l'azione "Compila" sul il progetto Libreria-Utente-Dinamica in modo da creare la nostra libreria dinamica Libreria-Utente-Dinamica.dll

N.B. Non effettuare mai né l'azione "Esegui" né l'azione "Compila & Esegui" in quanto OVVIAMENTE mancando di main si riceverà questo warning



Come usare una libreria dinamica in DEV C++

- 1 Creare un Nuovo Progetto di tipo Console Application
- 2 Assicurarsi che sia selezionato LO STESSO LINGUAGGIO scelto per la libreria (nel nostro caso C)
- 3 Dare un nome al nuovo progetto (nel nostro caso ancora Uso-Libreria-Utente)
- 4 Salvarlo nella cartella desiderata

  N.B. nel nostro caso per semplicità NELLA MEDESIMA DIRECTORY dove è posizionato il progetto Libreria-Utente-Dinamica

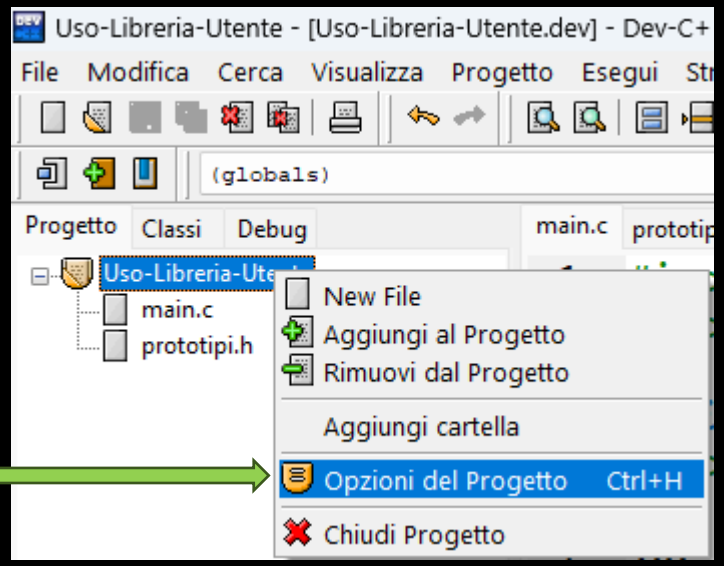
- 5 Aggiungere al progetto Uso-Libreria-Utente il file prototipi.h (SOLO se non è stato precedentemente incluso nel progetto della libreria dinamica)
- 6 Utilizzare nel file main.c del progetto le chiamate alla procedura SommaP ed alla funzion e SommaF

Come usare una libreria dinamica in DEV C++

- 7 Aggiungere nella sezione Linker del tab Parametri contenuto nelle Opzioni del Progetto il file creato in precedenza contenente la nostra Libreria-Utente-Dinamica.dll

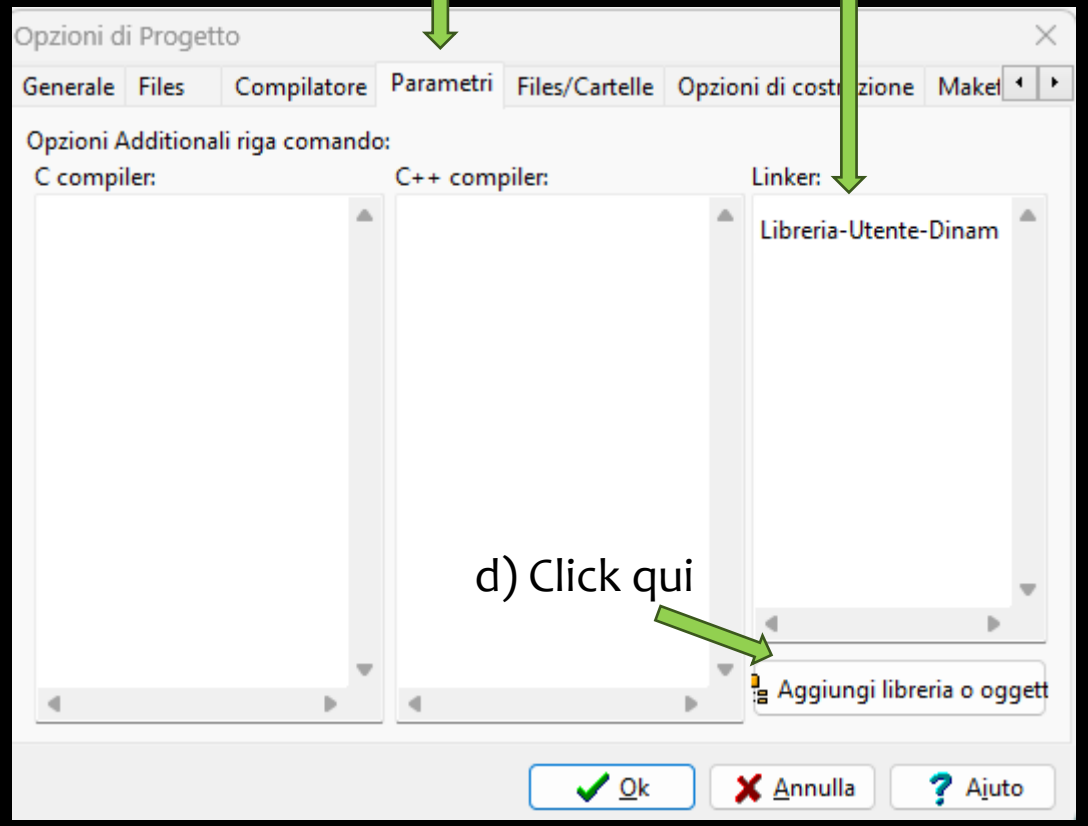
a) Tasto dx

b) Click qui



c) Scelta Tab

e) Aggiunta OK



d) Click qui

Come usare una libreria dinamica in DEV C++

- 8 Effettua l'azione "Compila" sul progetto Uso-Libreria-Utente avente come main.c un listato che invochi sia la procedura sia la funzione presenti nella libreria (come nel nostro esempio)

N.B. Se il file è stato già incluso nella libreria non va scritto

```c
#include <stdio.h>
#include <stdlib.h>

/* include file prototipi funzioni libreria utente */
#include "prototipi.h"

int main(int argc, char *argv[])
{
/* Dichiarazione variabili di input */
int a, b;
/* Dichiarazione variabili di output */
int s;

printf ("Inserire il valore di a: ");
scanf("%d", &a);
printf ("Inserire il valore di b: ");
scanf("%d", &b);

/* chiamata a procedura SommaP */
SommaP (a, b, &s);
printf("La somma calcolata con la PROCEDURA SommaP e': %d \n", s);

/* chiamata a funzione SommaF */
s = SommaF (a, b);
printf("La somma calcolata con la FUNZIONE SommaF e': %d \n", s);

return 0;
}
```

Chiamata alla PROCEDURA SommaP presente nella nostra libreria

Chiamata alla FUNZIONE SommaF presente nella nostra libreria