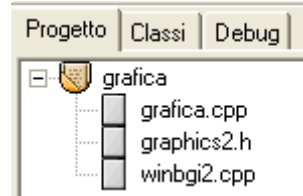


Graficare in ambiente Dev-C++

Bisogna includere nel progetto i due file:

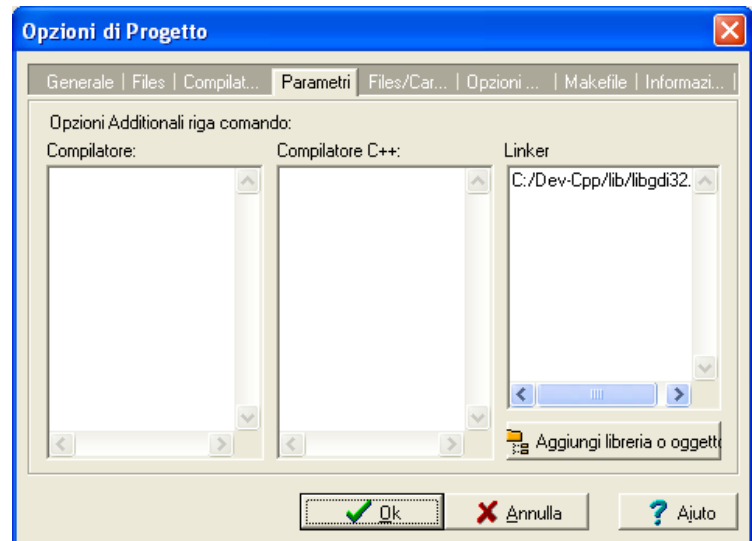
graphics2.h e **winbgi2.cpp**

copiandoli all'interno della cartella di lavoro oppure tramite la funzione AGGIUNGI AL PROGETTO



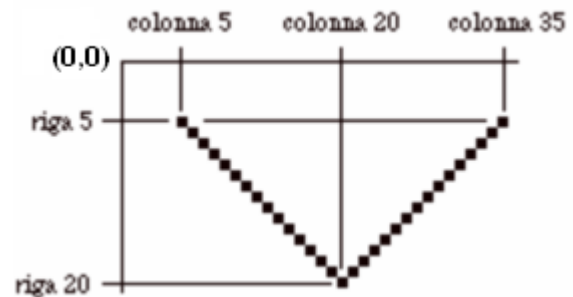
NB:

è importante aggiungere, alle librerie da linkare, il file **libgdi32.a** (presente in "DEV-CPP\lib") tramite il pannello relativo alle **Opzioni di Progetto** --> scheda **Parametri**



Si noti che per visualizzare pixel per pixel un'immagine sullo schermo, l'**origine** degli assi cartesiani rappresentati è impostato **in alto a sinistra** (si veda in figura il punto di coordinate $x=0, y=0$) e l'asse delle ordinate è orientato verso il basso.

Ad esempio nel caso della "V" tracciata in figura si possono tracciare le coordinate dei punti: $(5,5)$, $(6,6)$, ..., $(20,20)$, $(21,19)$, $(22,18)$, ..., $(34,6)$ e $(35,5)$



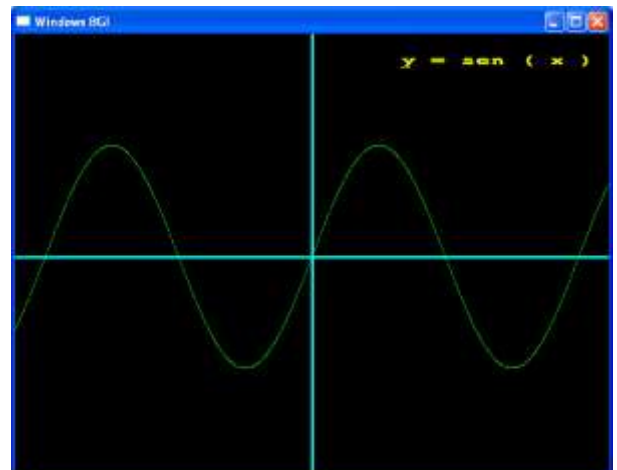
Codice di esempio

```
// disegno di sinusoide
```

```
#include <cmath>
```

```
#include "graphics2.h"
```

```
/* inclusione di un file header posizionato nella cartella  
corrente che definisce il prototipo delle funzioni grafiche */
```



```

int main()
{
    float x, y;
    int GraphDriver = DETECT, // rileva la più alta risoluzione possibile tipicamente
                                // VGA (640 x 480) ed equivale a GraphDriver=0
        GraphMode;           // auto-detected cioè GraphMode=0 tipicamente VGAHI

    int Xpixel= 640, Ypixel=480;
    initgraph (&GraphDriver, &GraphMode, "", Xpixel, Ypixel); // Start Window

    setcolor(CYAN); // il colore impostato per gli assi


    float x1, y1, x2, y2;
                                // traccio asse x
    x1 = 0;
    y1 = Ypixel / 2;
    x2 = Xpixel;
    y2 = y1;
    setlinestyle(SOLID_LINE, 0, 3);

    line (x1, y1, x2, y2); // line(int x1, int y1, int x2, int y2) ..... warning passando float
                                // con casting --> line ((int) x1, (int) y1, (int) x2, (int) y2);
                                // traccio asse y
    x1 = Xpixel / 2;
    y1 = 0;
    x2 = x1;
    y2 = Ypixel ;
    setlinestyle(SOLID_LINE, 0, 3);

    line (x1, y1, x2, y2); // line(int x1, int y1, int x2, int y2) ..... warning passsando float
                                // con casting --> line ((int) x1, (int) y1, (int) x2, (int) y2);

    // dimensiono area grafica
    // si potrebbero acquisire da tastiera ymax e xmax
    float ymax=4.0;
    float xmax=14.0;
    float unitaX = Xpixel / xmax;
    float unitaY = Ypixel / ymax;

    for (x=-7;x<7;x=x+0.01)
    {
        y=sin(x);
        putpixel(x*unitaX+320,-y*unitaY+240,2); // ..... warning passando float
    } // con casting --> putpixel((int) (x*unitaX) +320,-(int) (y*unitaY) +240,2);

    setcolor(YELLOW); // il colore impostato per la scritta
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 2);
    outtextxy(400, 20, " y = sen ( x ) "); // per visualizzare tale scritta è necessario
                                // non sostituire con chiamata a sistema la gestione di attesa
    getch(); // Wait for a key. (When main function ends, the window will close)
                                // nb: si può forzare la chiusura della GUI con pressione sulla tipica icona 
                                // forzando il ritorno all'IDE con analogia icona nella finestra console
} //end of main()

```