

# Guida a CustomGraphics

3<sup>a</sup>CET

Anno Scolastico 2022-2023



# Indice

0.1	Introduzione . . . . .	4
<b>1</b>	<b>Istruzioni</b>	<b>5</b>
1.1	GraphicScanfXY . . . . .	5
1.2	Assi . . . . .	6
1.3	TracciaPixel . . . . .	8
1.4	TracciaRetta . . . . .	11
1.5	TracciaRettaVert . . . . .	13
1.6	TracciaPunto . . . . .	15
1.7	TracciaSegmento . . . . .	17
1.8	TracciaCerchio . . . . .	20
1.9	TracciaParabola . . . . .	23
1.10	ScriviRetta . . . . .	26
1.11	ScriviRettaVert . . . . .	28
1.12	ScriviCerchio . . . . .	30
1.13	ScriviParabola . . . . .	32
1.14	ScriviEllisse . . . . .	34
1.15	RettaInt . . . . .	36

## 0.1 Introduzione

CustomGraphics è una libreria che integra diverse istruzioni per la programmazione grafica. E' presente all'interno del template "Pino10".

# Capitolo 1

## Istruzioni

### 1.1 GraphicScanfXY

L'istruzione "graphicscanfxy" viene usata per acquisire un dato da parte dell'utente. Questo valore, che può essere di tipo int, di tipo float o di tipo char, viene poi inserito all'interno di una variabile dichiarata in precedenza. In un ambiente di programmazione grafica non è possibile usare il comando "scanf", che viene invece usato in ambienti testuali.

La sintassi è la seguente:

```
graphicScanfxy(x, y, "formato", &var);
```

- x: indica la coordinata x del punto nel quale l'utente inserirà un valore;
- y: indica la coordinata y del punto nel quale l'utente inserirà un valore;
- "formato": indica il formato della variabile che si vuole acquisire ("%f" = float "%d" = int);
- var: il nome della variabile nella quale inserire il valore acquisito;
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

## 1.2 Assi

L'istruzione "assi" viene usata per visualizzare sullo schermo un diagramma cartesiano, da usare come base per tracciare grafici e diagrammi.

La sintassi è la seguente:

`assi(rx, ry, center);`

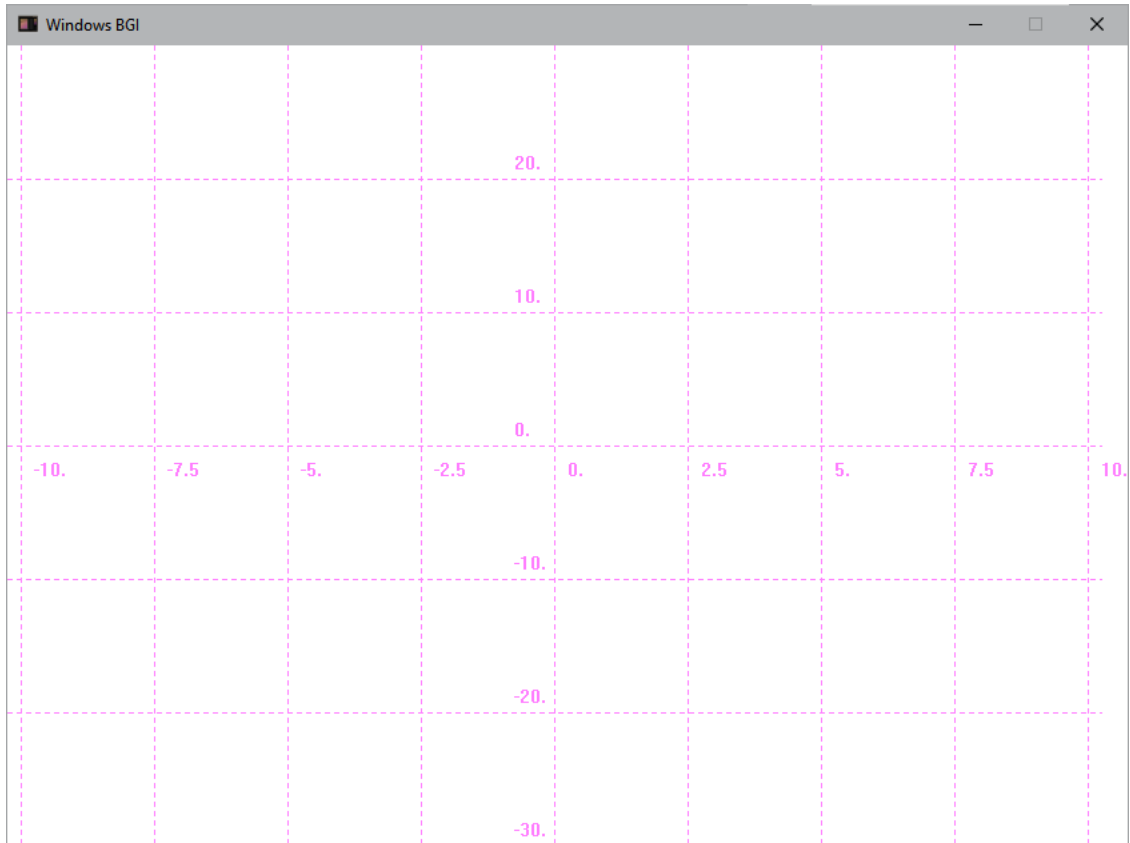
- rx: il numero di parti in cui è diviso l'asse delle ascisse (con rx=40 si ottiene un grafico che va da x=-20 a x=20);
- ry: il numero di parti in cui è diviso l'asse delle ordinate, in maniera analoga ad rx;
- center: indica la tipologia di grafico da disegnare: center=1 corrisponde ad un grafico con l'origine al centro della finestra, qualunque altro numero genera un grafico in cui l'origine è situata nell'angolo in basso a sinistra.

Questo programma genera un piano cartesiano che spazia da -10 a 10 nelle ascisse e da -30 a 30 nelle ordinate, con l'origine posta al centro della finestra.

Codice:

```
1 {  
2   initwindow(850, 610);  
3   assi(20, 60, 1);  
4   getch();  
5 }
```

Output:



### 1.3 TracciaPixel

L'istruzione "tracciapixel" viene usata per visualizzare un pixel, di posizione e colore a scelta, all'interno della finestra.

La sintassi è la seguente:

```
tracciapixel(x, y, rx, ry, col, center);
```

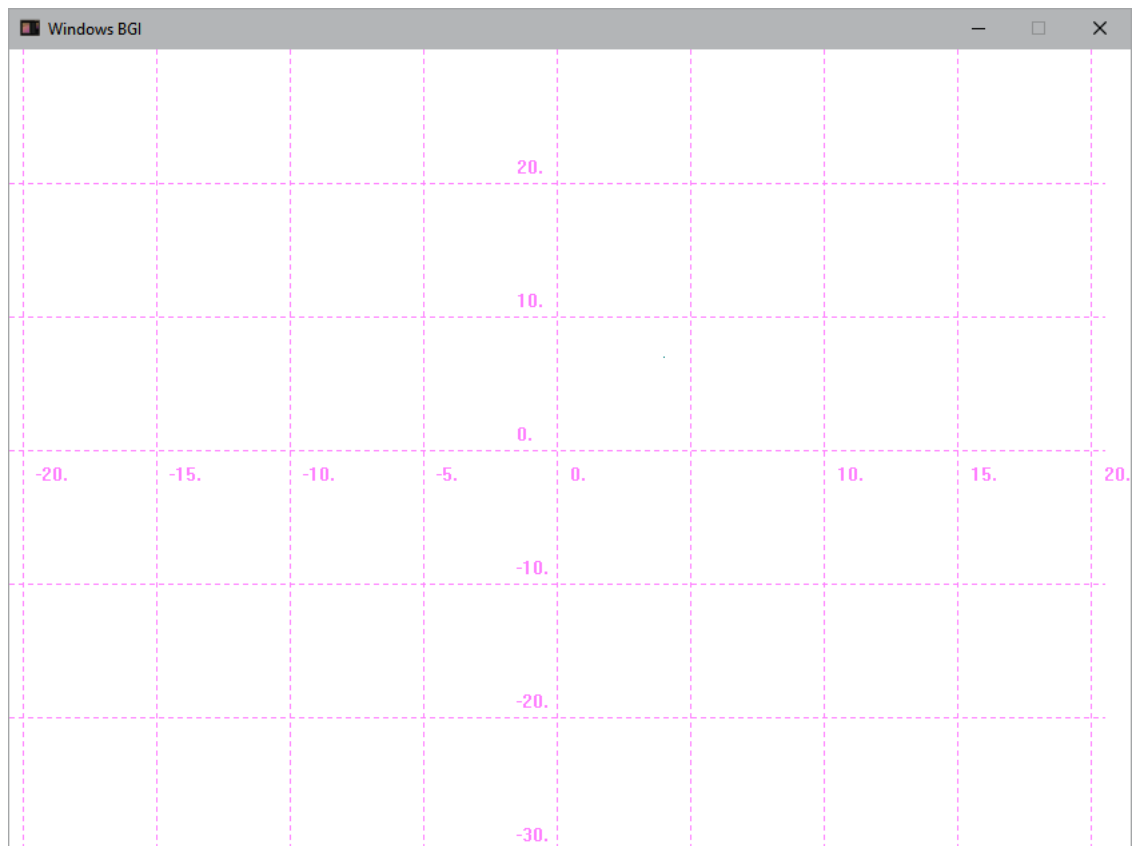
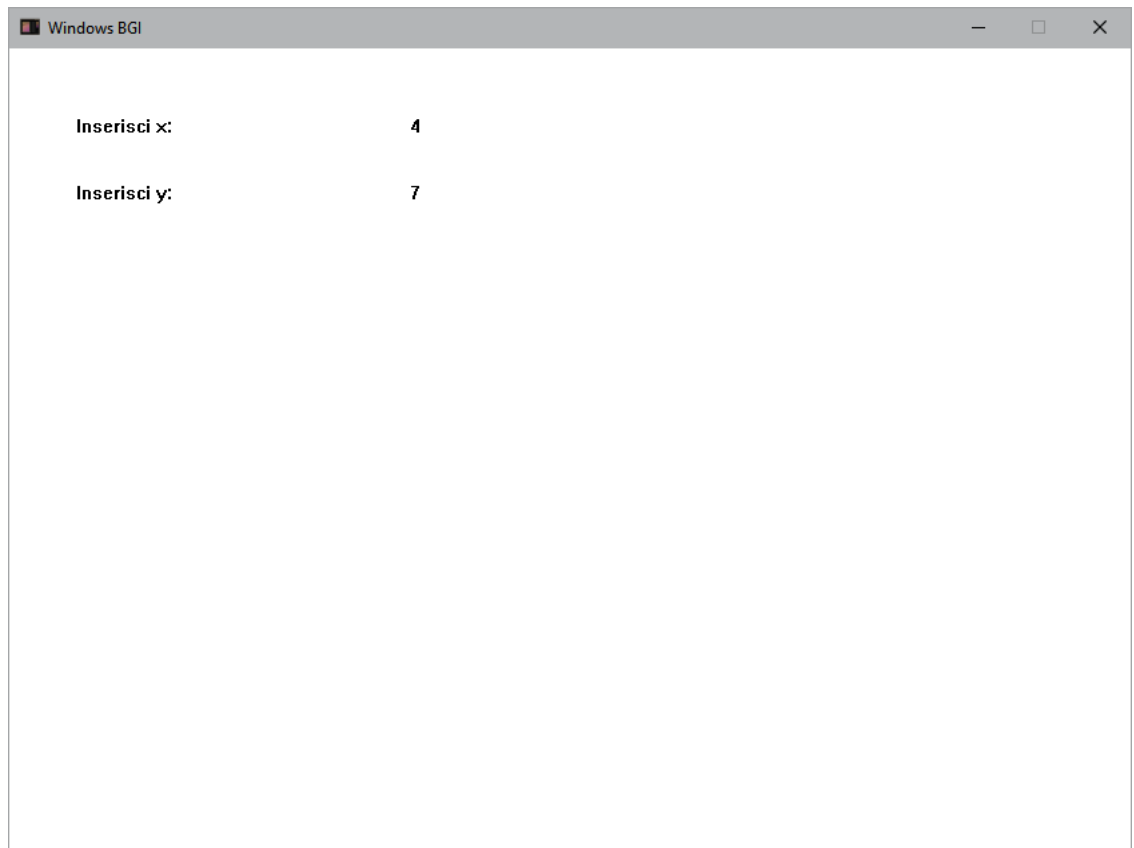
- x: indica la coordinata x del punto che si vuole tracciare sullo schermo;
- y: coordinata y del punto che si vuole tracciare sullo schermo;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del pixel.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma colora un singolo pixel su un diagramma cartesiano, dopo aver ottenuto le coordinate di esso tramite l'uso di "graphicScanfxy". Inoltre usa "cleardevice" per "ripulire" lo schermo. E' difficile da notare, ma a coordinate (4,7) è presente un pixel colorato di rosso.



Codice e output:

```
1      {
2      initwindow(850, 610);
3      float x, y;
4
5      outtextxy(50, 50, "Inserisci x:");
6      graphicScanfxy(300, 50, "%f", &x);
7      outtextxy(50, 100, "Inserisci y:");
8      graphicScanfxy(300, 100, "%f", &y);
9      cleardevice();
10
11     int rx, ry;
12     rx = 40;
13     ry = 60;
14     assi(rx, ry, 1);
15
16     tracciapixel(x,y,rx,ry,4,1);
17     getch();
18 }
```



## 1.4 TracciaRetta

L'istruzione "tracciaretta" viene usata per creare una linea retta di pixel, di colore a scelta, per visualizzare una retta all'interno della finestra. Viene usata insieme al comando "assi" per creare grafici.

La sintassi è la seguente:

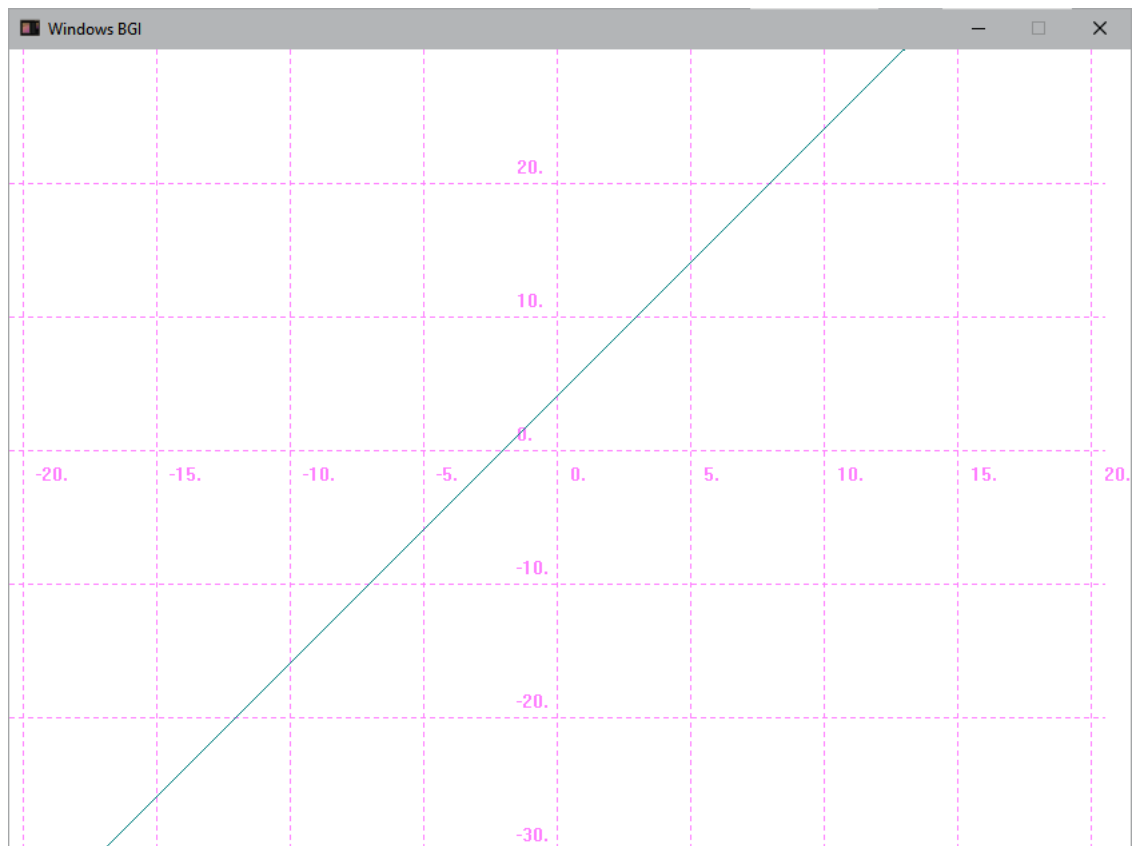
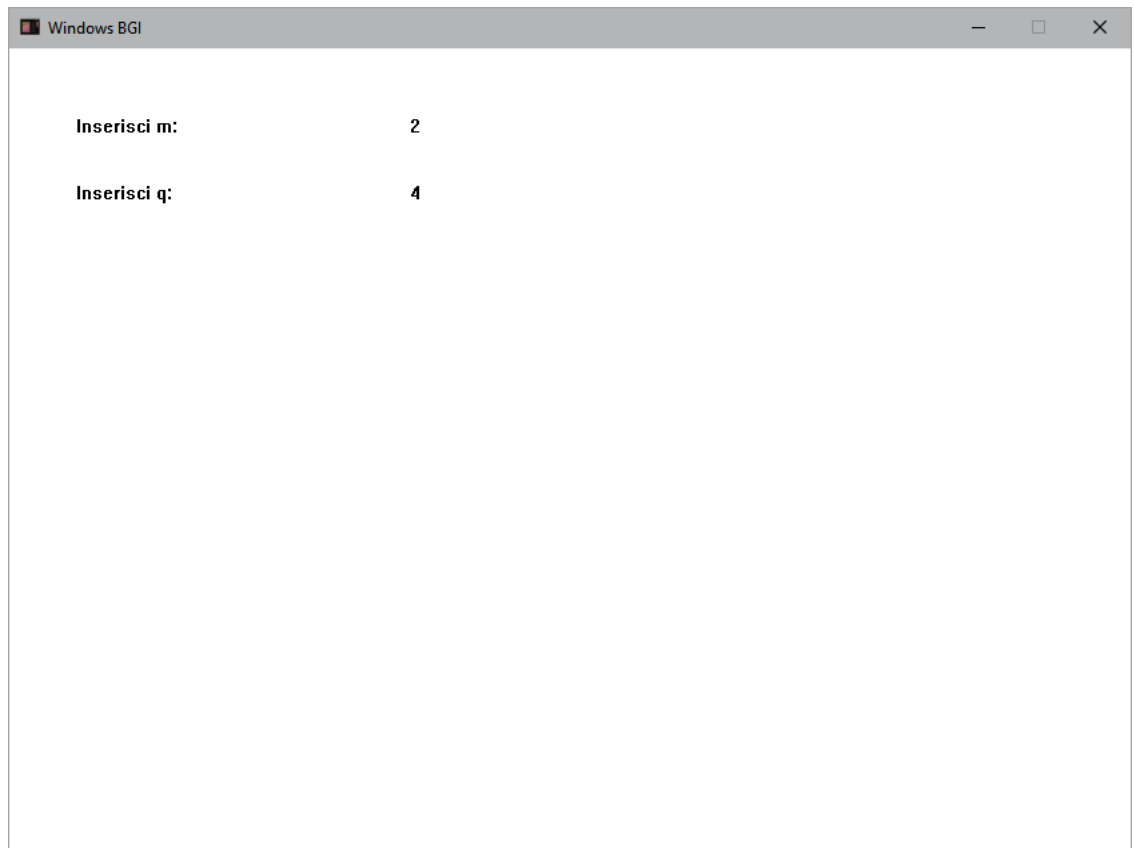
`tracciaretta(m, q, rx, ry, col, center);`

- m: indica il coefficiente angolare della retta da rappresentare
- q: indica la quota della retta da rappresentare;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del pixel.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna una retta su un diagramma cartesiano, dopo aver ottenuto quota e coefficiente angolare tramite l'uso di "graphicScanfxy". Inoltre usa "cleardevice" per "ripulire" lo schermo.

Codice e output:

```
1      {
2          initwindow(850, 610);
3          float m, q;
4          int rx, ry;
5          rx = 40;
6          ry = 60;
7
8          outtextxy(50, 50, "Inserisci m:");
9          graphicScanfxy(300, 50, "%f", &m);
10         outtextxy(50, 100, "Inserisci q:");
11         graphicScanfxy(300, 100, "%f", &q);
12
13         cleardevice();
14         assi(rx, ry, 1);
15         tracciaretta(m, q, rx, ry, 12, 1);
16         getch();
17     }
```



## 1.5 TracciaRettaVert

L'istruzione "tracciarettavert" viene usata per creare una linea verticale di pixel, di colore a scelta, per visualizzare una retta all'interno della finestra. Viene usata insieme al comando "assi" per creare grafici.

La sintassi è la seguente:

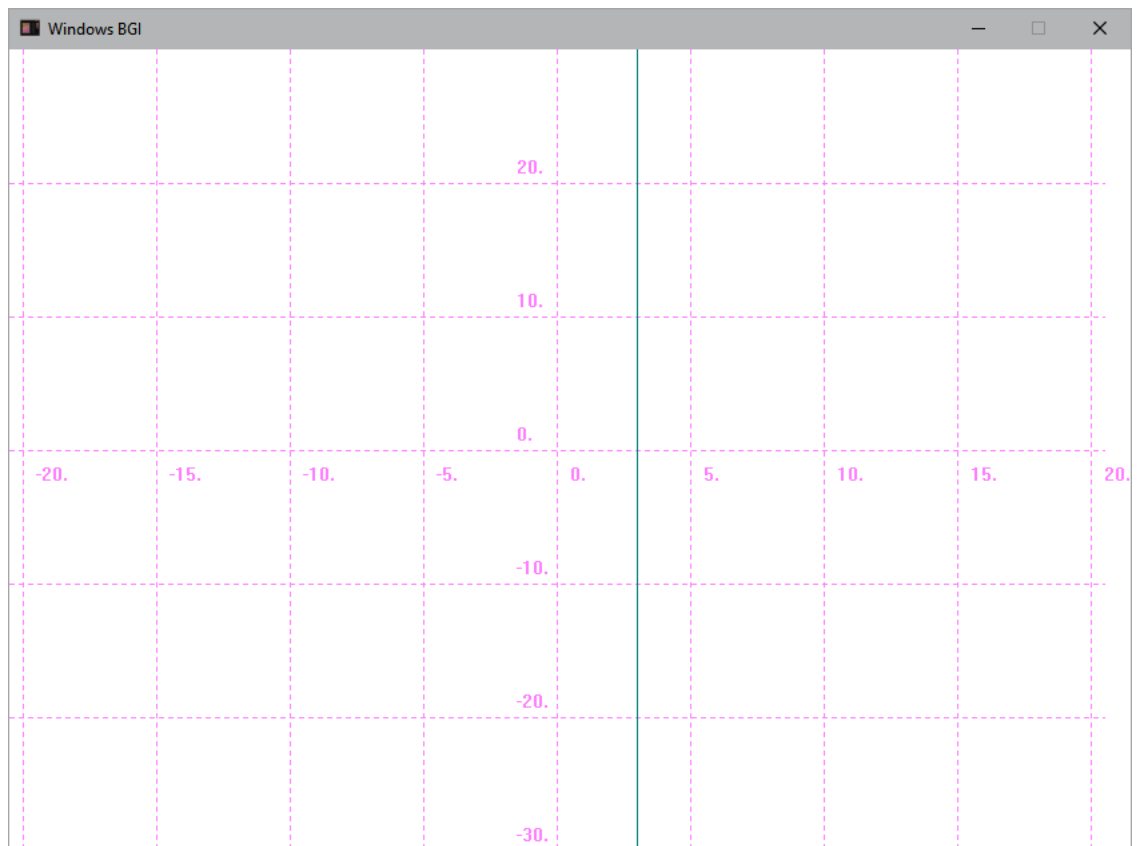
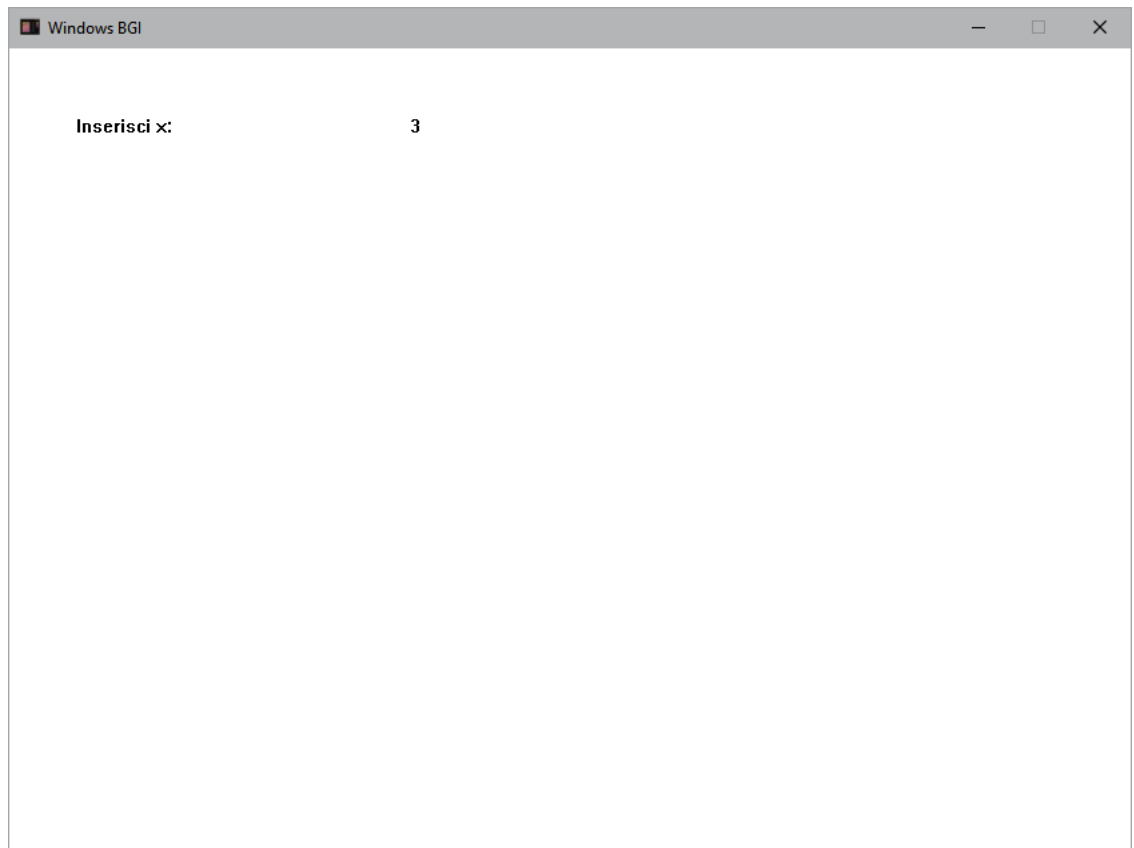
`tracciarettavert(x, rx, ry, col, center);`

- x: indica lo spostamento rispetto all'asse y della retta da rappresentare (Offset);
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del pixel.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna una retta verticale su un diagramma cartesiano, dopo aver ottenuto l'offset tramite l'uso di "graphicScanfxy". Inoltre usa "cleardevice" per "ripulire" lo schermo.

Codice e output:

```
1      {
2      initwindow(850, 610);
3      float x;
4
5      outtextxy(50, 50, "Inserisci x:");
6      graphicScanfxy(300, 50, "%f", &x);
7
8      cleardevice();
9
10     int rx, ry;
11     rx = 40;
12     ry = 60;
13     assi(rx, ry, 1);
14
15     tracciarettavert(x, rx, ry, 12, 1);
16     getch();
17 }
```



## 1.6 TracciaPunto

L'istruzione "tracciapunto" viene usata per disegnare un punto, di colore a scelta, all'interno della finestra. Il punto disegnato non è un singolo pixel, ma un piccolo cerchio. Viene spesso usata per segnare un'intersezione o per rappresentare graficamente l'input dell'utente.

La sintassi è la seguente:

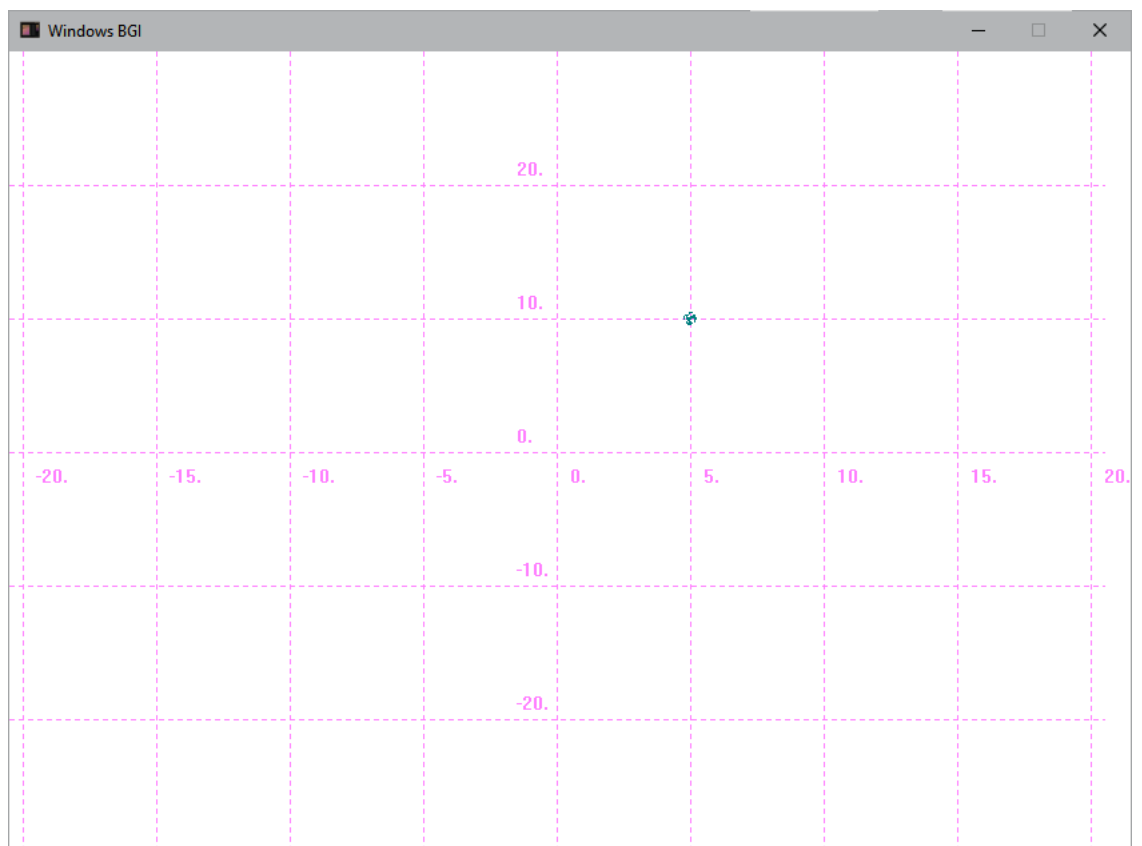
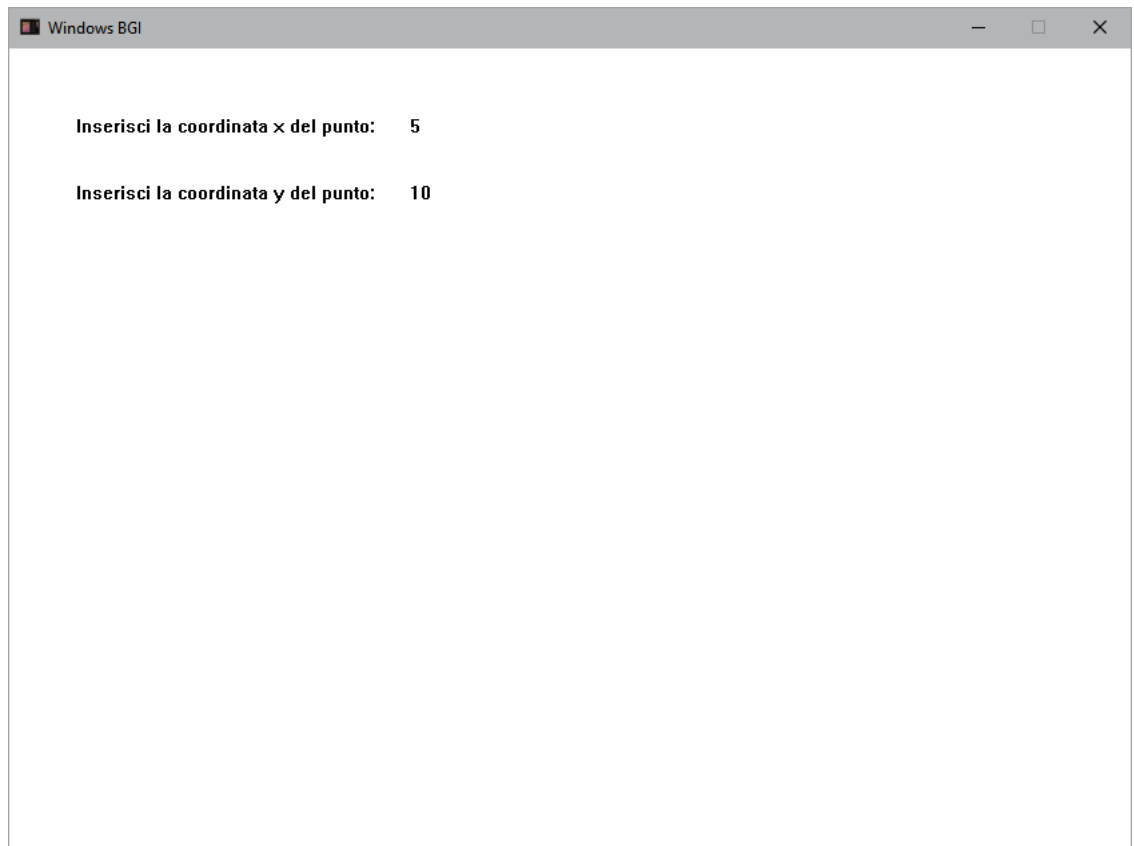
`tracciapunto(x, y, rx, ry, col, center);`

- x: indica la coordinata x del punto da rappresentare;
- y: indica la coordinata y del punto da rappresentare;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del pixel.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna un punto su un diagramma cartesiano, dopo aver ottenuto le coordinate tramite l'uso di "graphicScanfxy".

Codice e output:

```
1      {
2          initwindow(850, 610);
3          float x, y;
4
5          outtextxy(50, 50, "Inserisci la x del punto:");
6          graphicScanfxy(300, 50, "%f", &x);
7          outtextxy(50, 100, "Inserisci la y del punto:");
8          graphicScanfxy(300, 100, "%f", &y);
9          cleardevice();
10
11         int rx, ry;
12         rx = 40;
13         ry = 60;
14         assi(rx, ry, 1);
15         tracciapunto(x, y, rx, ry, 12, 1);
16         getch();
17     }
```





## 1.7 TracciaSegmento

L'istruzione "tracciasegmento" viene usata per tracciare un segmento, di colore a scelta, all'interno della finestra. Il segmento disegnato è quello che collega i punti A e B, inseriti nel comando. Può anche tracciare segmenti verticali.

La sintassi è la seguente:

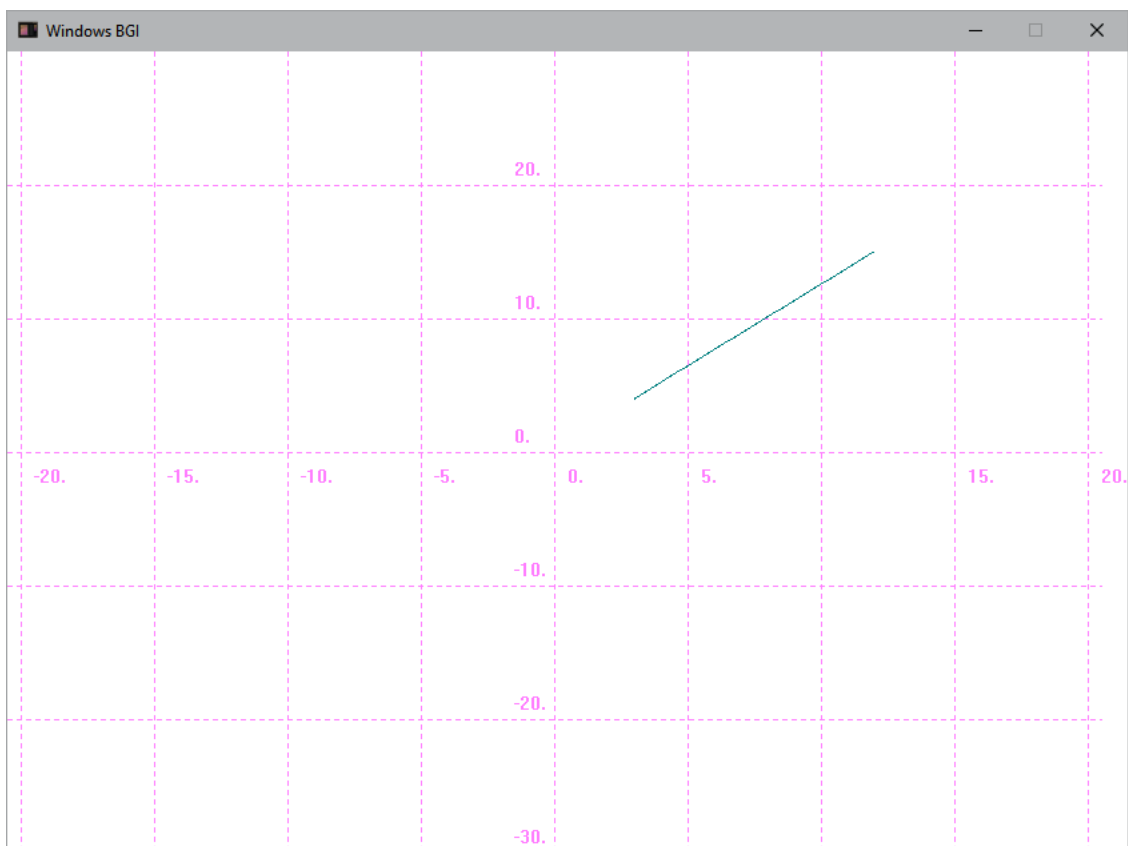
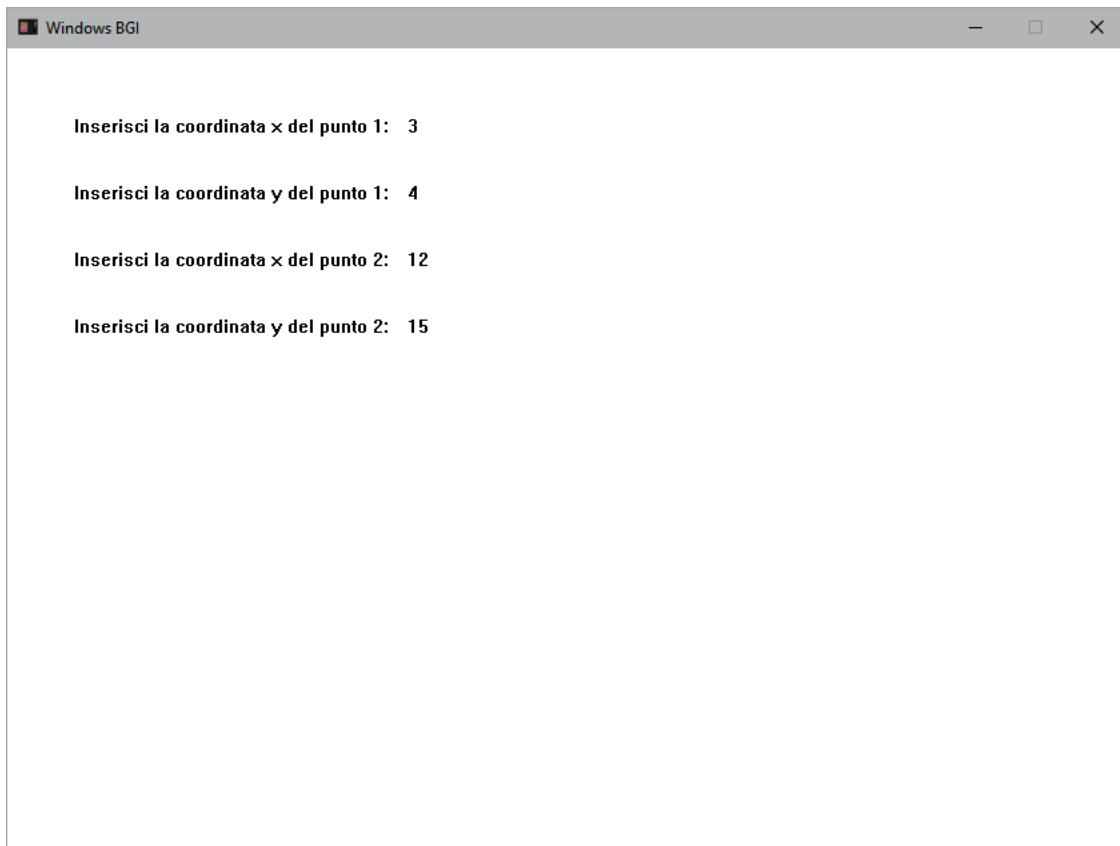
`tracciasegmento(x1, y1, x2, y2, rx, ry, col, center);`

- x1: indica la coordinata x del primo punto del segmento;
- y1: indica la coordinata y del primo punto del segmento;
- x2: indica la coordinata x del secondo punto del segmento;
- y2: indica la coordinata y del secondo punto del segmento;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del segmento.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna un segmento su un diagramma cartesiano, dopo aver ottenuto le coordinate tramite l'uso di "graphicScanfxy".

Codice e output:

```
1  {
2      initwindow(850, 610);
3      float x1, y1, x2, y2;
4
5      outtextxy(50, 50, "Inserisci la coord. x del punto 1:");
6      graphicScanfxy(300, 50, "%f", &x1);
7      outtextxy(50, 100, "Inserisci la coord. y del punto 1:");
8      graphicScanfxy(300, 100, "%f", &y1);
9      outtextxy(50, 150, "Inserisci la coord. x del punto 2:");
10     graphicScanfxy(300, 150, "%f", &x2);
11     outtextxy(50, 200, "Inserisci la coord. y del punto 2:");
12     graphicScanfxy(300, 200, "%f", &y2);
13     cleardevice();
14
15     int rx, ry;
16     rx = 40;
17     ry = 60;
18     assi(rx, ry, 1);
19
20     tracciasegmento(x1, y1, x2, y2, rx, ry, 12, 1);
21     getch();
22 }
```



## 1.8 TracciaCerchio

L'istruzione "tracciacerchio" viene usata per tracciare una circonferenza, di colore a scelta, all'interno della finestra. Quest'istruzione disegna una circonferenza data la sua equazione, mentre per disegnare un cerchio da raggio e coordinate del centro viene usato il comando "circle", compreso all'interno delle librerie standard del template "pino".

La sintassi è la seguente:

`tracciasegmento(a, b, c, rx, ry, col, center);`

- a: indica il primo coefficiente dell'equazione della circonferenza ( $ax^2$ );
- b: indica il secondo coefficiente dell'equazione della circonferenza ( $ax$ );
- c: indica il terzo coefficiente dell'equazione della circonferenza;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del segmento.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna una circonferenza su un diagramma cartesiano, dopo aver ottenuto le coordinate tramite l'uso di "graphicScanfxy".

Per far sì che la circonferenza risulti effettivamente circolare, rx ed ry devono essere proporzionati in maniera tale che un unità sull'asse x abbia lo stesso numero di pixel di un unità nell'asse y. Per fare ciò, nel programma, ho impostato  $rx = 40$  ed  $ry = 30$ .

Codice e output:

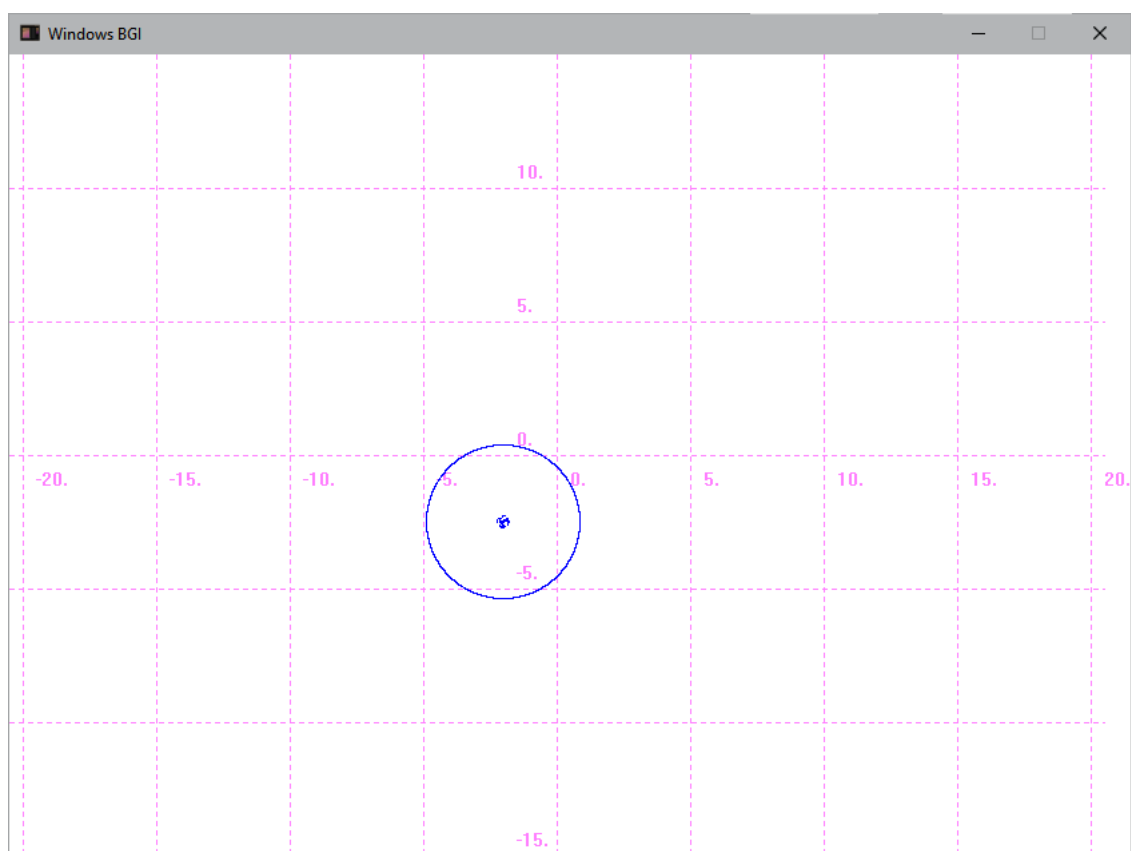
```
1      {
2          initwindow(850, 610);
3          float a,b,c;
4
5          outtextxy(50, 50, "Inserisci il coefficiente a:");
6          graphicScanfxy(300, 50, "%f", &a);
7          outtextxy(50, 100, "Inserisci il coefficiente b:");
8          graphicScanfxy(300, 100, "%f", &b);
9          outtextxy(50, 150, "Inserisci il coefficiente c:");
10         graphicScanfxy(300, 150, "%f", &c);
11         cleardevice();
12
13         int rx, ry;
14         rx = 40;
15         ry = 30;
16         assi(rx, ry, 1);
17
18         tracciacerchio(a, b, c, rx, ry, 14, 1);
19         getch();
20     }
```

Windows BGI

Inserisci il coefficiente a: 4

Inserisci il coefficiente b: 5

Inserisci il coefficiente c: 2



## 1.9 TracciaParabola

L'istruzione "tracciaparabola" viene usata per tracciare una parabola, di colore a scelta, all'interno della finestra. Quest'istruzione disegna una parabola data la sua equazione, che deve essere inserita in forma esplicita.

La sintassi è la seguente:

`tracciaparabola(a, b, c, rx, ry, col, center);`

- a: indica il primo coefficiente dell'equazione della parabola ( $ax^2$ );
- b: indica il secondo coefficiente dell'equazione della parabola ( $ax$ );
- c: indica il terzo coefficiente dell'equazione della parabola;
- rx: Il numero di parti in cui è diviso l'asse delle ascisse (Bisogna inserire lo stesso valore inserito nella funzione assi);
- ry: Il numero di parti in cui è diviso l'asse delle ordinate (Bisogna inserire lo stesso valore inserito nella funzione assi);
- col: indica il colore del segmento.
- center: indica il centro del diagramma cartesiano disegnato con la funzione assi (Bisogna inserire lo stesso valore inserito nella funzione assi);

Questo programma disegna una parabola su un diagramma cartesiano, dopo aver ottenuto le coordinate tramite l'uso di "graphicScanfxy".

Codice e output:

```
1  {
2      initwindow(850, 610);
3      float a,b,c;
4
5      outtextxy(50, 50, "Inserisci il coefficiente a:");
6      graphicScanfxy(300, 50, "%f", &a);
7      outtextxy(50, 100, "Inserisci il coefficiente b:");
8      graphicScanfxy(300, 100, "%f", &b);
9      outtextxy(50, 150, "Inserisci il coefficiente c:");
10     graphicScanfxy(300, 150, "%f", &c);
11
12     cleardevice();
13
14     int rx, ry;
15     rx = 40;
16     ry = 30;
17     assi(rx, ry, 1);
18
19     tracciaparabola(a, b, c, rx, ry, 14, 1);
20     getch();
21 }
```

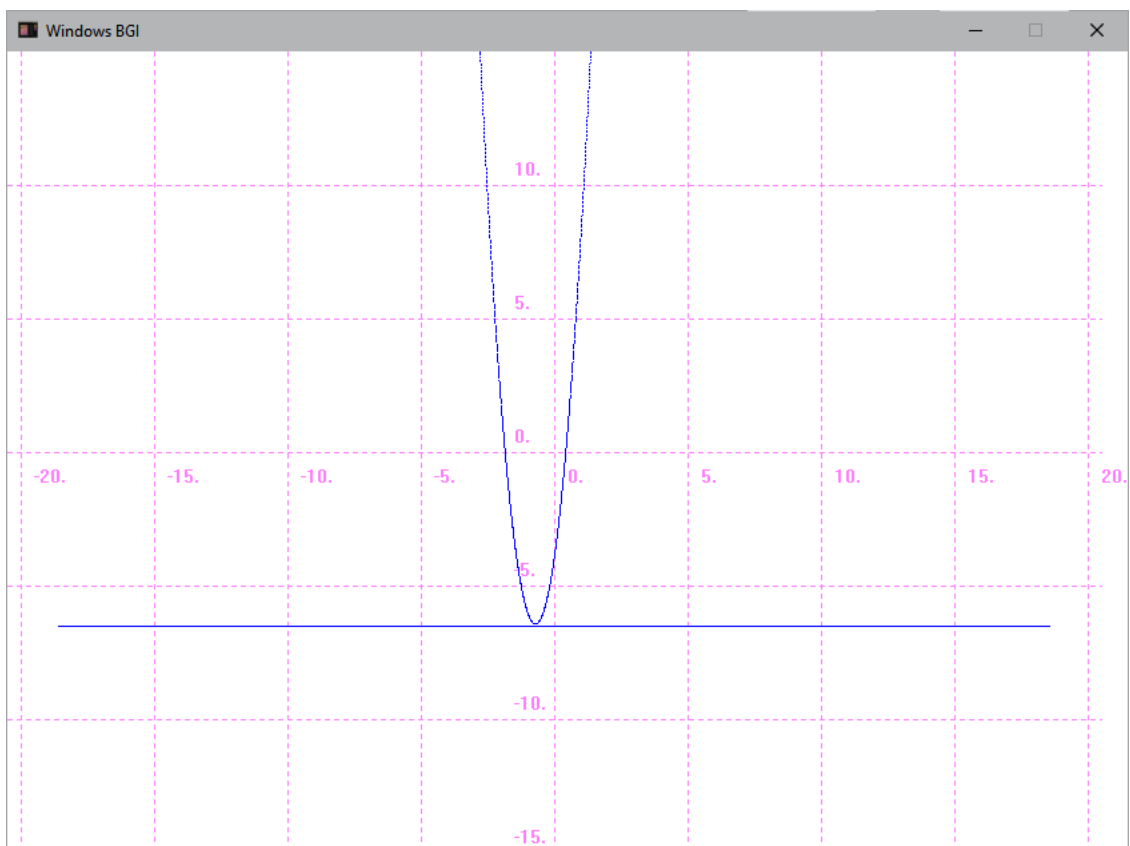


Windows BGI

Inserisci il coefficiente a: 5

Inserisci il coefficiente b: 7

Inserisci il coefficiente c: -4



## 1.10 ScriviRetta

L'istruzione "scriviretta" viene usata per scrivere a schermo l'equazione di una retta.

La sintassi è la seguente:

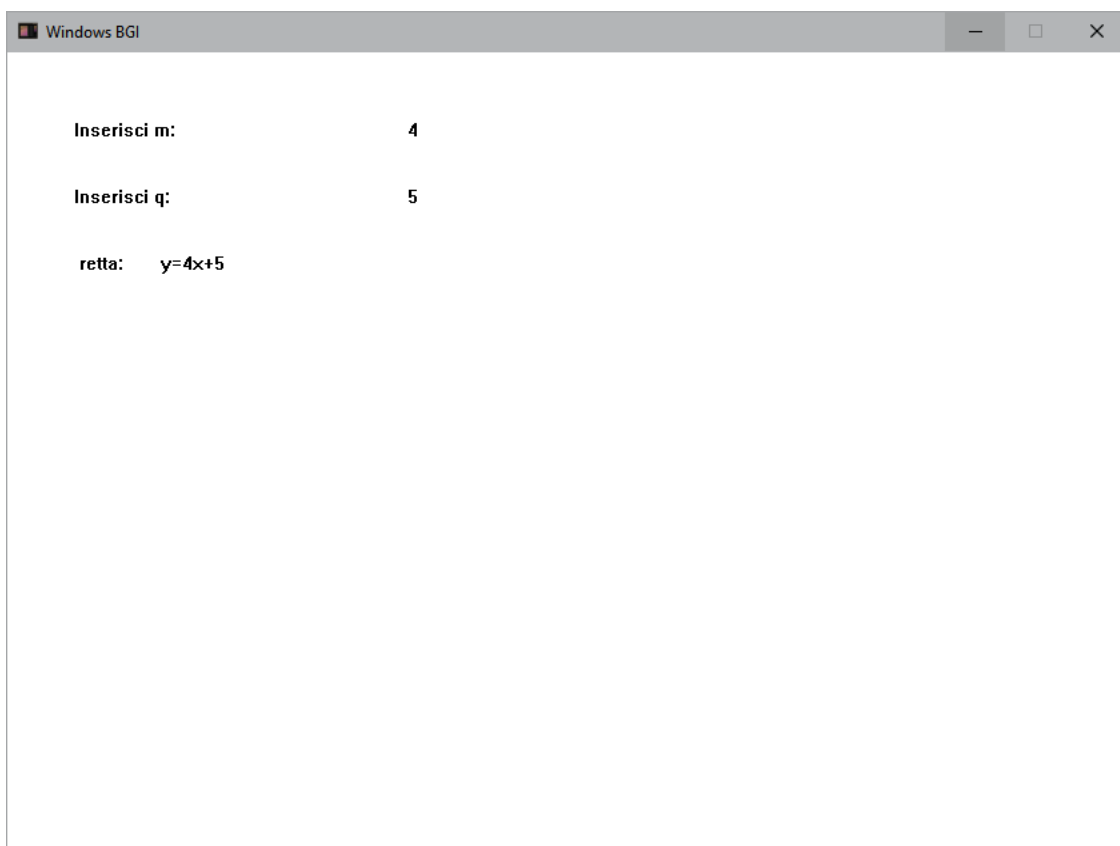
`scriviretta(m, q, y);`

- m: indica il coefficiente angolare della retta da scrivere;
- q: indica la quota della retta da scrivere (il valore y con x=0);
- y: la posizione del testo all'interno della pagina;

Questo programma scrive l'equazione della retta, usando dati inseriti dall'utente, tramite l'uso dei comandi `outtextxy` e `graphicScanfxy`.

Codice e output:

```
1      {  
2          initwindow(850, 610);  
3          float m, y, q;  
4          y = 150;  
5  
6          outtextxy(50, 50, "Inserisci m:");  
7          graphicScanfxy(300, 50, "%f", &m);  
8  
9          outtextxy(50, 100, "Inserisci q:");  
10         graphicScanfxy(300, 100, "%f", &q);  
11  
12         scriviretta(m, q, y)  
13         getch();  
14     }
```



## 1.11 ScriviRettaVert

L'istruzione "scrivirettavert" viene usata per scrivere a schermo l'equazione di una retta verticale.

La sintassi è la seguente:

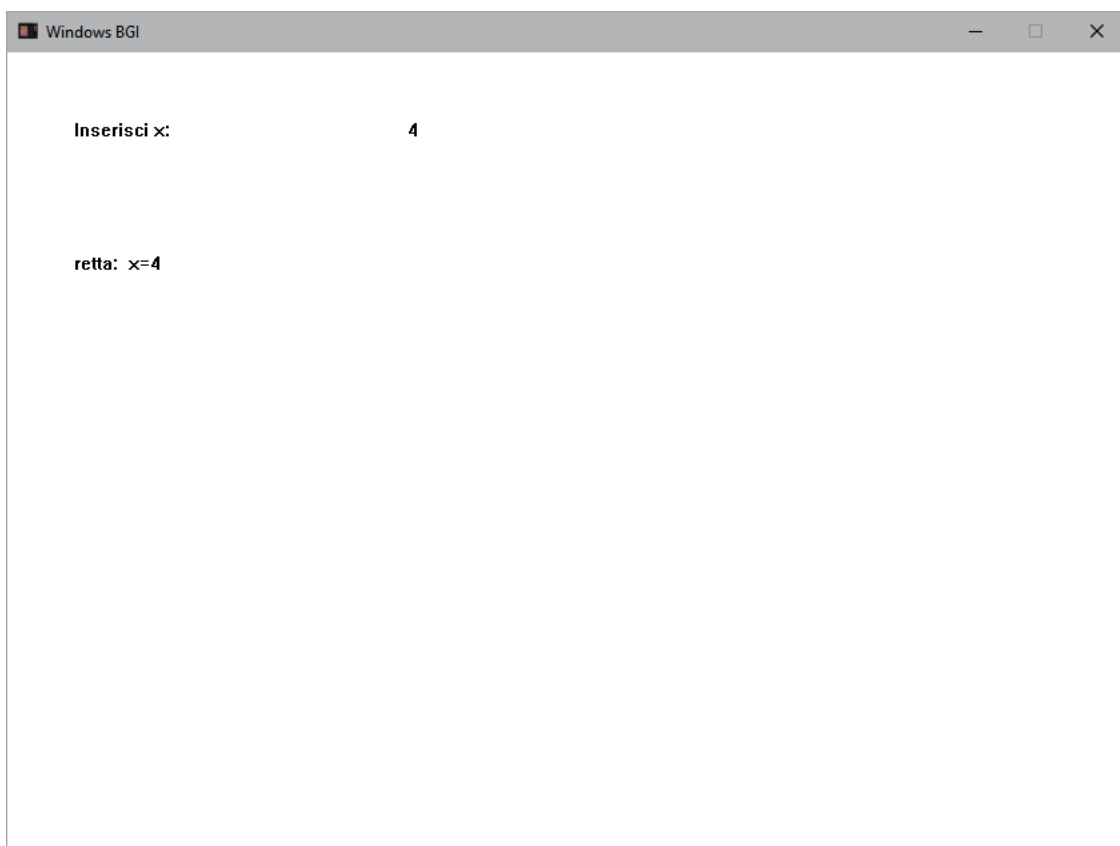
`scrivirettavert(x ,y);`

- x: indica il valore x della retta da scrivere;
- y: l'altezza del testo all'interno della finestra;

Questo programma scrive l'equazione di una retta verticale, usando dati inseriti dall'utente, tramite l'uso dei comandi `outtextxy` e `graphicScanfxy`.

Codice e output:

```
1      {  
2      initwindow(850, 610);  
3      float x, y;  
4      y = 150;  
5  
6      outtextxy(50, 50, "Inserisci x:");  
7      graphicScanfxy(300, 50, "%f", &x);  
8  
9      scrivirettavert(x, y);  
10     getch();  
11     }
```



## 1.12 ScriviCerchio

L'istruzione "scrivicerchio" viene usata per scrivere a schermo l'equazione di una circonferenza.

La sintassi è la seguente:

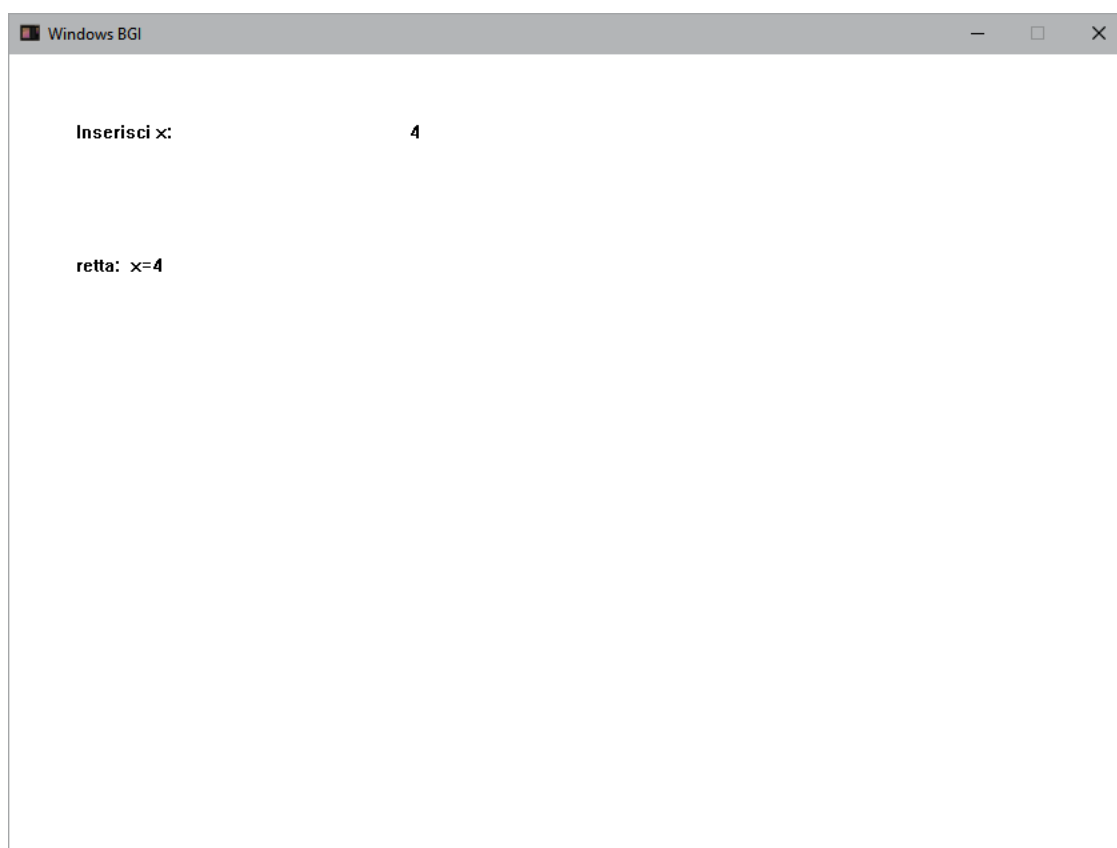
`scrivicerchio(a, b, c, y);`

- a: indica il primo coefficiente dell'equazione della circonferenza ( $ax^2$ );
- b: indica il secondo coefficiente dell'equazione della circonferenza ( $ax$ );
- c: indica il terzo coefficiente dell'equazione della circonferenza;
- y: la posizione del testo all'interno della pagina;

Questo programma scrive l'equazione di una circonferenza, usando dati inseriti dall'utente, tramite l'uso dei comandi `outtextxy` e `graphicScanfxy`.

Codice e output:

```
1      {  
2          initwindow(850, 610);  
3          float a, b, c, y;  
4          y = 250;  
5  
6          outtextxy(50, 50, "Inserisci a:");  
7          graphicScanfxy(300, 100, "%f", &a);  
8          outtextxy(50, 100, "Inserisci b:");  
9          graphicScanfxy(300, 150, "%f", &b);  
10         outtextxy(50, 50, "Inserisci c:");  
11         graphicScanfxy(300, 150, "%f", &c);  
12  
13         scrivicerchio(a, b, c, y);  
14         getch();  
15     }
```



## 1.13 ScriviParabola

L'istruzione "scriviparabola" viene usata per scrivere a schermo l'equazione di una parabola.

La sintassi è la seguente:

`scriviparabola(a, b, c, y);`

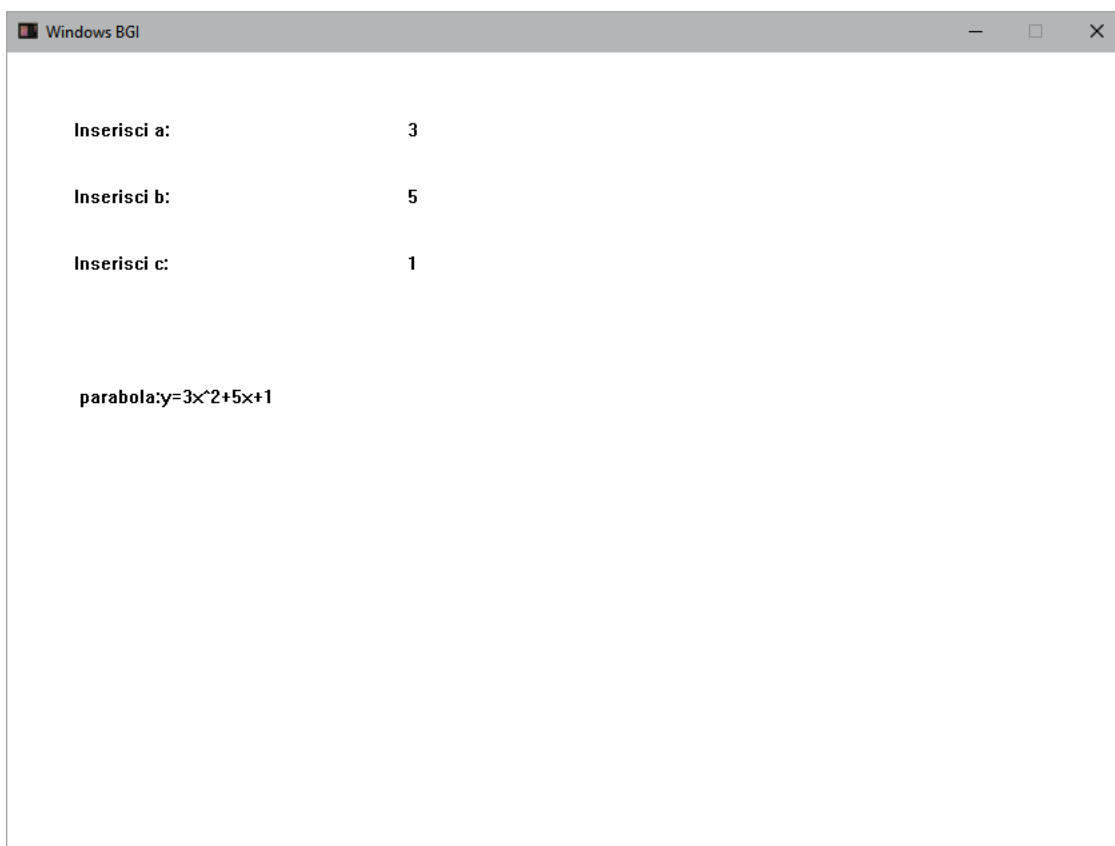
- a: indica il primo coefficiente dell'equazione della parabola ( $ax^2$ );
- b: indica il secondo coefficiente dell'equazione della parabola ( $ax$ );
- c: indica il terzo coefficiente dell'equazione della parabola;
- y: la posizione del testo all'interno della pagina;

Questo programma scrive l'equazione di una parabola, usando dati inseriti dall'utente, tramite l'uso dei comandi `outtextxy` e `graphicScanfxy`.

Codice e output:

```
1      {
2          initwindow(850, 610);
3          float a, b, c, y;
4          y = 250;
5
6          outtextxy(50, 50, "Inserisci a:");
7          graphicScanfxy(300, 50, "%f", &a);
8          outtextxy(50, 100, "Inserisci b:");
9          graphicScanfxy(300, 100, "%f", &b);
10         outtextxy(50, 150, "Inserisci c:");
11         graphicScanfxy(300, 150, "%f", &c);
12
13         scriviparabola(a, b, c, y);
14         getch();
15     }
```





## 1.14 ScriviEllisse

L'istruzione "scriviellisse" viene usata per scrivere a schermo l'equazione di un'ellisse.

La sintassi è la seguente:

`scriviparabola(a, b, y);`

- a: indica il primo coefficiente dell'equazione dell'ellisse ( $ax^2$ );
- b: indica il secondo coefficiente dell'equazione dell'ellisse ( $ax$ );
- y: la posizione del testo all'interno della pagina;

Questo programma scrive l'equazione di una parabola, usando dati inseriti dall'utente, tramite l'uso dei comandi `outtextxy` e `graphicScanfxy`. Codice: e output:

```
1      {  
2      initwindow(850, 610);  
3      float a, b, y;  
4      y = 200;  
5  
6      outtextxy(50, 50, "Inserisci a:");  
7      graphicScanfxy(300, 50, "%f", &a);  
8      outtextxy(50, 100, "Inserisci b:");  
9      graphicScanfxy(300, 100, "%f", &b);  
10  
11     scrivieellisse(a, b, y);  
12     getch();  
13 }
```

