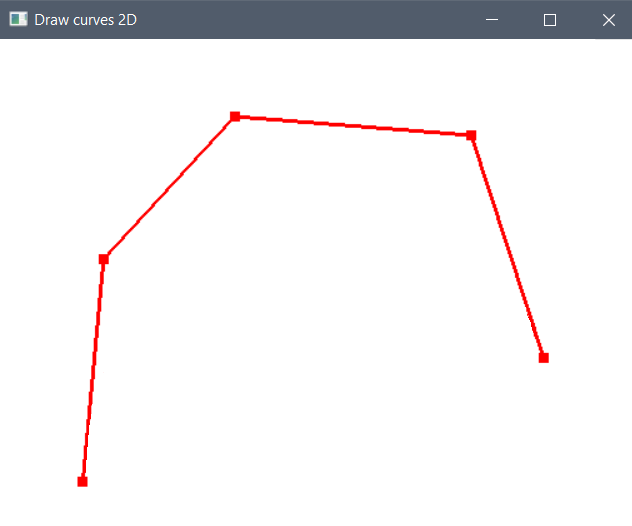
LAB 1 – DISEGNO DI CURVE DI BÉZIER

# **0. Test sulle funzionalità esistenti**

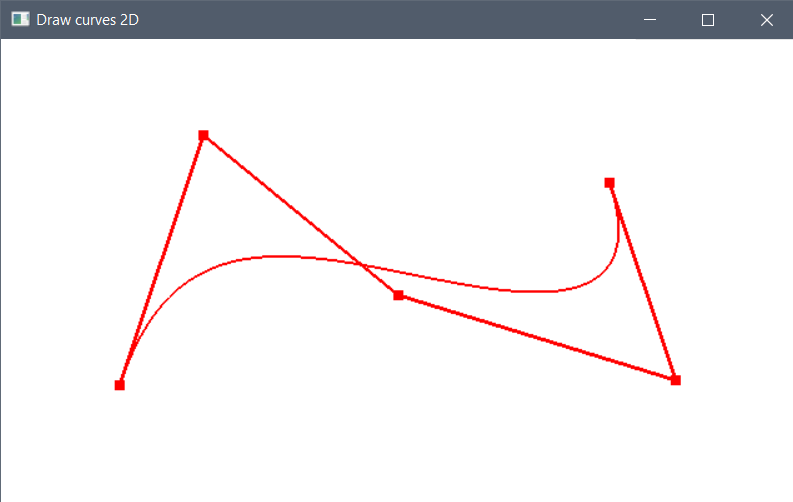
La prima parte del laboratorio (punti 1. e 2. della consegna) ha previsto l’utilizzo delle funzionalità già presenti nel codice di partenza fornito. In particolare, sono stati testati i controlli da keyboard (‘f’ ed ‘l’ rimuovono il primo e l’ultimo punto dalla lista di punti, rispettivamente) e da mouse (il left button aggiunge un punto alla lista, fino ad un massimo di 64 nuovi punti). Il programma, inoltre, disegna i punti e la poligonale che li connette.

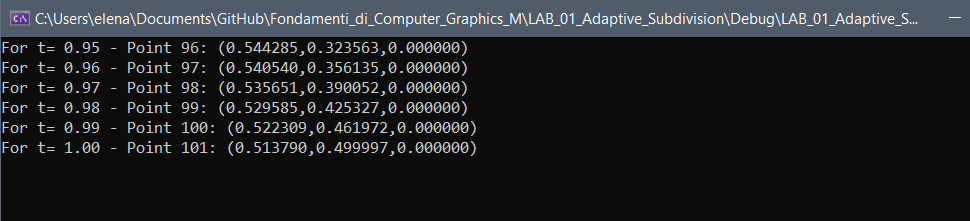
Più approfonditamente, attraverso il codice, è stato possibile osservare l’utilizzo delle funzioni di callback OpenGL GLUT per la cattura di eventi di mouse e tastiera e la relativa associazione dei metodi da eseguire all’avvenire degli eventi.



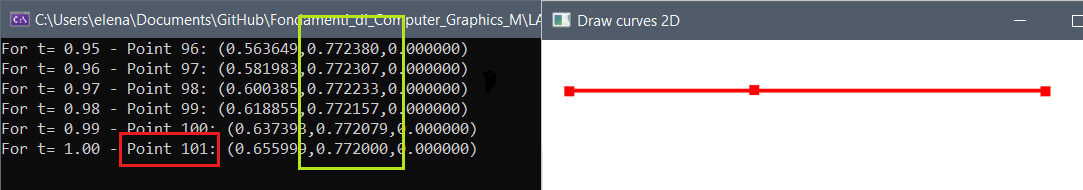
# **1. Utilizzo di de Casteljau**

Il punto 3 della consegna ha richiesto di “*disegnare la curva di Bézier a partire dai punti di controllo inseriti, utilizzando l’algoritmo di de Casteljau*”. Per questa prima richiesta, è stato sufficiente applicare la **discretizzazione uniforme** per cui si discretizza l’intervallo parametrico , in *N* punti equidistanti, per ognuno di questi punti si applica de Casteljau ottenendo il punto sulla curva di Bézier e la curva mostrata a schermo è la poligonale che connette questi ultimi. Ovviamente, più alto è *N* più definita sarà la curva, ma più costoso sarà generarla.





Nell’esempio è stata utilizzata una variazione di *t* di *0.01*, che ha portato alla generazione di punti tra cui anche il primo e l’ultimo punto della lista della poligonale iniziale (corrispondenti a e ). Questa soluzione permette di avere una precisione abbastanza alta nel rendering della curva, ma porta come svantaggio il fatto che il numero di punti sia prefissato, anche nel caso in cui la curva sia in realtà una semplice retta.



**2. Suddivisione adattiva**

Il punto 4 della consegna ha richiesto di scegliere tra due alternative:

1. l’applicazione della suddivisione adattiva
2. gestione di curve di Bézier a tratti della relativa discontinuità.

È stato scelto il punto a), quindi il “disegno di una curva di Bézier mediante algoritmo ottimizzato basato sulla suddivisione adattiva”. La suddivisione adattiva mira ad ottimizzare l’algoritmo precedente “adattando” il numero di punti (e quindi di segmenti) in base alle caratteristiche della curva. Si prevede che ad ogni iterazione, venga esaminata la