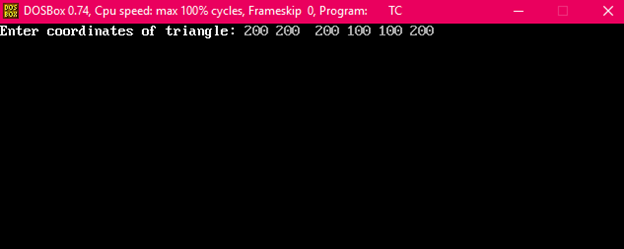
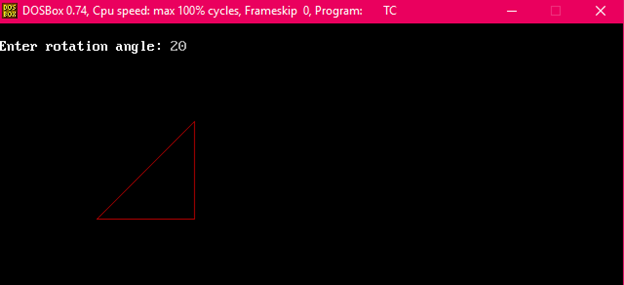
Program to rotate a Triangle:

1. #include<stdio.h>
2. #include<graphics.h>
3. #include<math.h>
4. main()
5. {
6. intgd=0,gm,x1,y1,x2,y2,x3,y3;
7. **double** s,c, angle;
8. initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
9. setcolor(RED);
10. printf("Enter coordinates of triangle: ");
11. scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2, &x3, &y3);
12. setbkcolor(WHITE);
13. cleardevice();
14. line(x1,y1,x2,y2);
15. line(x2,y2, x3,y3);
16. line(x3, y3, x1, y1);
17. getch();
18. setbkcolor(BLACK);
19. printf("Enter rotation angle: ");
20. scanf("%lf", &angle);
21. setbkcolor(WHITE);
22. c = cos(angle \*M\_PI/180);
23. s = sin(angle \*M\_PI/180);
24. x1 = floor(x1 \* c + y1 \* s);
25. y1 = floor(-x1 \* s + y1 \* c);
26. x2 = floor(x2 \* c + y2 \* s);
27. y2 = floor(-x2 \* s + y2 \* c);
28. x3 = floor(x3 \* c + y3 \* s);
29. y3 = floor(-x3 \* s + y3 \* c);
30. cleardevice();
31. line(x1, y1 ,x2, y2);
32. line(x2,y2, x3,y3);
33. line(x3, y3, x1, y1);
34. getch();
35. closegraph();
36. **return** 0;
37. }

**Output:**

**Before rotation**

**After rotation**

