Q.1 a.20   (=15+5)

Q.2

```
#include <simplecpp>
main_program{
        turtleSim();
        repeat(4)
        {
                forward(10) ;
                penUp();
                forward(20);
                penDown();
        }
        cout << "thanks";
        wait(10);
}
```

Q.3
   A.50
   B.West

Q.4 ii) 610 (= 500+100+10)

Q.5 i) 1 (the turtle goes around the same pentagon so we see just one pentagon in the output)

Q.6 A and C

Q.7

$\{(x, y) \mid x=$even and $y=$even$\} \cup \{(6480,1) , (2160,3) , (1296,5) , (720,9) , (432,15) , (16,405) , (240,27) , (144,45) , (48,135) , (80,81)\}$

Basically x*y=6480 and x is even and y is odd. So this part
repeat(y){   right(180); forward(6480/x); } // 6480/x = y
left(180);
forward(y);
becomes equivalent to 0 displacement.

Q.8   (x,y)=(5pq,-10p)

Q.9

```
#include <simplecpp>
main_program{
        turtleSim();
        cout << "What is the area";
        int area;
        cin >> area;
        repeat(3)
        {
    forward(sqrt(area*4/sqrt(3)));
    left(120);
    }
wait(10);
}
```

Q.10 Answer: 2*360/(2*PI)

Q.11

No. It is not possible to trace a circle which is perfectly smooth. It is because a perfectly smooth circle will have infinite number of sides. Even by using repeat command it will need infinite number of iterations to complete the circle which is impossible.

Q.12
    circum/360

Q.13

   A. (180-arcsine(45.0/75))

   B. (arccosine(-60.0/75))   or   (180-(arccosine(60.0/75)))

   C. (180-arctangent(45.0/60)) or   (arctangent2(45,-60))

   D. 75 (= sqrt(45^2+60^2)=75)

Q.14

A. 2*side*cosine(36);

Look at the triangle which is made up by 2 sides of pentagon and one side of star. Now by some construction in triangle and known values(angle, side of pentagon) you have to find this.

B. (180-36)

Q. 15

Correct program is :

```
#include<simplecpp>
main_program{
turtleSim();
int side;
      side = 360 ;
      repeat(side/6){
              forward(7);
       left(360.0/(side/6));
              repeat(side){
              forward(1);
              right(360.0/side); //semicolon missing
              }
      } // closing brace missing
       wait(20);
       cout << "Done" ; // opening quotation missing , wrong cout syntax
}
```