(1) $1+\frac{1}{3^2}+\frac{1}{5^2}+\frac{1}{7^2}+\cdots\cdots=\frac{\pi^2}{8}$

(2) $\frac{1}{2}+\frac{1}{1\times3}-\frac{1}{3\times5}+\frac{1}{5\times7}-\frac{1}{7\times9}+\cdots=\frac{\pi}{4}$

(3) $1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-+\cdots\cdots=\frac{\pi}{4}$

(1)

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main()
{
floati,y,pi,sum=0.0;
      for(i=1.0;i<=10000.0;i++){
            y=1.0/(float)pow(2.0*i-1.0,2.0);
            sum+=y;
      }
      pi=sqrt(sum*8.0);              //same as pi=pow(sum*8.0,1.0/2.0);
      printf("pi=%.6f \n",pi);       //shown as %.6f ,but test in %.20f
}
```

(2)

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main()
{
floati,x,y,z,pi,sum=0.5;
      for(i=1.0;i<=100.0;i++){
            x=pow(-1.0,i+1);
            y=1.0/((2*i-1)*(2*(i+1)-1));
            z=x*y;
            sum+=z;
      }
      pi=sum*4.0;
      printf("pi=%.6f \n",pi);
}
```

(3)

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void main()
{
floati,x,y,z,pi,sum=0.0;
      for(i=1.0;i<=1000000.0;i++){
            x=pow(-1.0,i+1);
            y=1.0/(2*i-1);
            z=x*y;
            sum+=z;
      }
      pi=sum*4.0;
      printf("pi=%.6f \n",pi);
}
```

Results：

(1) when i<=10 ,pi=3.109625

when i<=100 ,pi=3.138408

when i<=1000 ,pi=3.141273

when i<=10000 ,pi=3.141459

when i<=100000 ,pi=3.1414597 (same as i<=10000)

The value convergence at 3.14145970344543457000 when i<=10000

(2) when i<=10 ,pi=3.137078

when i<=100 ,pi=3.141544

when i<=1000 ,pi=3.141593

when i<=10000 ,pi=3.141593 (same as i<=1000)

The value convergence at 3.14159321784973145000 when i<=1000

(3) when i<=10 ,pi=3.041839

when i<=100 ,pi=3.131593

when i<=1000 ,pi=3.140593

when i<=10000 ,pi=3.141499

when i<=100000 ,pi=3.141585

when i<=1000000 ,pi=3.1415954

The value convergence at 3.14159536361694336000 when i<=1000000

Conclusion：

三個算式都將收斂於圓周率，但是，收斂的速度不同，目前測試之結果爲（２）＞（１）＞（３）。