一、

1. B
2. A
3. B
4. C　　//此题没看懂
5. D
6. B
7. C
8. D
9. C
10. A

二、

1.

1

2.

18

3.

3 5 3 5　　这题没看懂

4.

5

5.

10

三、

1.

(1)

```
int i,j,sum
```

(2)

```
1000
```

(3)

```
sum=1
```

(4)

```
i%j==0
```

(5)

```
sum==i
```

2.

(6)

```
string *s //不确定
```

(7)

```
*s
```

(8)

```
rp>lp
```

(9)

```
s++
```

(10)

```
lp!=p?0
```

四、

1.

```
#include<stdio.h>
float p(int n, float x){
 if (n==0)
 return 0;
 else if(n==1)
 return x;
 else
 return ((2*n-1)*x*p(n-1,x)-(n-1)*p(n-2,x))/n;
}
int main() {
 float rs = p(4,1.5);
 printf("%f\n",rs);
 return 0;
}
```

2.

```
#include <stdio.h>
#include <math.h>
int main() {
 double x=1,y=0,i=3,f=-1;
 while (fabs(x-y)>1e-6){
 y=x;
 x=x+f*(1/i);
 f=-1*f;
 i+=2;
```

```
 }
 printf("%lf\n", x);
 return 0;
}
```

3.

```
#include <stdio.h>
int main() {
 float len =0,high=1000;
 for (int i = 0; i < 10; ++i) {
 len+=high;
 high=high/2;
 len+=high;
 }
 len-=high;
 printf("第十次落地时,共经过%fm\n", len);
 printf("第十次反弹%fm\n", high);
 return 0;
}
```

4.

```
#include <stdio.h>
const int num = 9; //最高坐标
int main() {
 //m数据数组，a,b目前塞数的坐标，n欲塞的数，flag 1-往左上方塞数 2-往右下方塞数
 int m[num+1][num+1],a=num,b=0,n=2,flag=1;
 m[num][0]=1; //先将1塞进
 while(n<=(num+1)*(num+1)) {
 if (flag == 1){
 if (b==0 && a-1>=0){ //碰左壁
  a=a-1;
  m[a][b] = n++;
  flag=2;
 } else if(a==0){ //碰上壁
  b=b+1;
  m[a][b]=n++;
  flag=2;
 } else{
  a=a-1;
  b=b-1;
  m[a][b]=n++;
 }
```

```
} else if(flag == 2){
if(b==num){ //碰右壁
a=a-1;
m[a][b]=n++;
flag = 1;
} else if(a==num){ //碰下壁
b=b+1;
m[a][b]=n++;
flag=1;
} else{
b=b+1;
a=a+1;
m[a][b]=n++;
}
}
}
//打印数组
for (int i = 0; i < num+1; ++i) {
for (int j = 0; j < num+1; ++j) {
printf("%d\t", m[i][j]);
}
printf("\n");
}
return 0;
}
```