一、

1. A
2. D
3. B
4. A
5. C
6. D
7. D
8. D
9. B
10. C

二、

1. 1 0 5
2. 72
3. 60
4. a*(b*(c*(d+exp(x))+y)+log(z))
5. (x*x+y*y)>(a*a)&&(x*x+y*y)<(b*b)
6. 1 2 0

三、

1. 39
2. 5,6,5,6,10
3. 1146
4. 18
5. 10,16

四、

1.

(1) #define N 100

(2) s=0

(3)a[i]

(4)continue

(5)count++

2.

(6)return

(7)(struct node *)malloc(sizeof(struct node))

(8)input()

(9)p!=NULL

(10)p=p->next

3.

(11)*(s1 + i - n + n1 - 1)

(12)c==1?tchar-10+'0':tchar+'0'

(13)return result

(14)num = ladd(num1, num2)

(15)free(num)

五、

1.

```c
#include <stdio.h>
float fun(int n, float x){
 if(n==0)
 return 1;
 return x * fun(n - 1, x);
}
int main(){
 int n;
 float x;
  printf("请输入n,x:\n");
  scanf("%d%f", &n, &x);
  printf("pow(%d,%f)=%f\n", n, x, fun(n, x));
  return 0;
 }
```

2.

```c
#include <stdio.h>
int a[53];
void fun(int n){
 int t=n;
 while (n<=52){
 if (a[n]==0)
 a[n]=1;
 else
 a[n]=0;
  n+=t;
   }
 }
```

```
int main(){
 int i,count=0;
 for ( i= 1; i <= 52; ++i)
 a[i]=1;

 for (i = 2; i <= 52; ++i)
 fun(i);

 printf("正面朝上的牌序号数:\n");
 for (i = 1; i <= 52; ++i) {
 if(a[i]==1){
 count++;
 printf("%d\t",i+1);
 }
 }
 printf("\n正面朝上一共%d张牌\n", count);
 return 0;
}
```

3.

```
#include <stdio.h>
#include <math.h>
int verify(int n){
 for (int i = 2; i <= (int)sqrt(n); ++i) {
 if (n%i==0)
 return 0;
 }
 return 1;
}
int main(){
 int count=0,x=0;
 FILE *file;
 if((file=fopen("prime.txt","w"))==NULL){
 printf("打开文件失败\n");
 return 0;
 }
 for (int i = 2; i <=300 ; ++i) {
 if (verify(i)){
 fprintf(file, "%d\n", i);
 if(i+2<=300&&verify(i+2)){
 count++;
```

```
  x=i;
  }
  }
  }
  fclose(file);
  printf("[2,300]内一共%d对双胞胎数\n最大一对是%d,%d\n", count, x, x + 2);
  return 0;
}
```