

一、

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

二、

- 6 5
- 字母或下划线
- float
- 43
- 1
- $a*(b*(c*(d+x*x)+y)+z)$
- 3
- 1 0

三、

1. $i=30$
2. 1#2#3#4#
3. 3456789012
4. !yppah
5. 60,2,30,2,15,3,5,5,

四、

1、

- (1) $i < N$
- (2) $i < N$
- (3) $< N-i$
- (4) $a[j] > a[j+1]$
- (5) $a[j] = a[j+1]$

2、

(6)&n

(7)m%10

(8)m/10

(9)m>0

(10)r(m)

3、

(11)a[i]+b[i]

(12)a[i+1]

(13)c++

(14)*cp

(15)c-1

五、

1、

```
1  #include <stdio.h>
2  int main(void) {
3      int n,x;
4      int T(int n, int x, FILE *file);
5      printf("请输入n,x:\n");
6      scanf("%d%d", &n, &x);
7      FILE *file;
8      if((file=fopen("tdat.txt","w"))==NULL){
9          printf("打开文件失败\n");
10         return 0;
11     }
12     for (int i = 0; i <= n; ++i) {
13         fprintf(file, "T%d(%d)=%d\n", i, x, T(i, x, file));
14     }
15
16     return 0;
17 }
18 int T(int n, int x, FILE *file){
19     int t=0;
20     if (n==0){
21         t=1;
22         return t;
23     } else if(n==1){
24         t=x;
25         return t;
```

```

26 }
27 t = 2 * x * T(n - 1, x, file) - T(n - 2, x, file);
28 return t;
29 }

```

2、

```

1 #include <stdio.h>
2 int main(void) {
3     float a,b,res1;
4     int res2,flag=1;
5     char m;
6     printf("请输入两个操作数和一个双目算数运算符:\n");
7     scanf("%f%f%c", &a, &b, &m);
8     switch (m){
9     case '+':
10        res1=a+b;
11        break;
12     case '-':
13        res1 = a-b;
14        break;
15     case '*':
16        res1 = a * b;
17        break;
18     case '/':
19        res1 = a/b;
20        break;
21     case '%':
22        res2 = (int) a % (int) b;
23        flag=0;
24    }
25    if (flag)
26        printf("%f\n", res1);
27    else
28        printf("%d\n", res2);
29
30    return 0;
31 }

```

3、

```

1 #include <stdio.h>
2 #include <stdlib.h>
3

```

```

4  /**
5   * 验证列最小
6   * @param m 二维数组头地址
7   * @param i 需验证数所在行数
8   * @param j 需验证数所在列数
9   * @param n 阶数
10  * @return
11  */
12 int verify(int *m,int i,int j,int n){
13     int t = *(m+i*n+j);
14     int flag=0;
15     for (int k = 0; k < n; ++k) {
16         if (*(m+k*n+j)<=t && k!=i)
17             return flag;
18     }
19     flag=1;
20     return flag;
21 }
22 int main(void) {
23     int n;
24     printf("请输入方阵阶数:\n");
25     scanf("%d", &n);
26     int *m = (int *) malloc(sizeof(int) * n * n);
27     printf("请输入方阵中的元素，由左至右，由上至下:\n");
28     for (int i = 0; i < n*n; ++i) {
29         scanf("%d", m + i);
30     }
31
32     //寻找每行中的行最小值
33     int j=0,min=0;
34     for (int k = 0; k < n; ++k) {
35         min = *(m + k * n);
36         j=0;
37         for (int l = 1; l < n; ++l) {
38             if(*(m+k*n+l)<min){
39                 min = *(m+k*n+l);
40                 j=l;
41             }
42         }
43         //验证行最小是否为列最小

```

```
44  if (verify(m, k, j, n)) {  
45  printf("%d(%d, %d)\n", min, k+1, j+1);  
46  }  
47  }  
48  
49  return 0;  
50 }
```