C语言-最后一课

S.IST@XMU

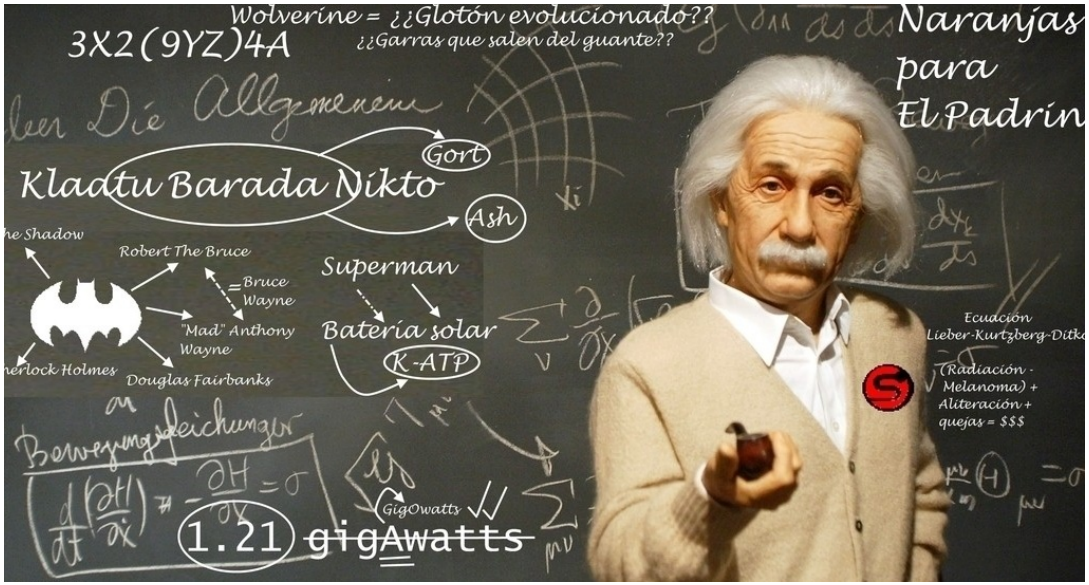
复习回顾

- 上次课的内容：
  - 二进制文件读写
  - rewind
  - fseek和ftell
  - 文件的随机访问
  - 如何看待上机考试：F1排位赛

考前当学霸，幸福一寒假！

大学教育是什么

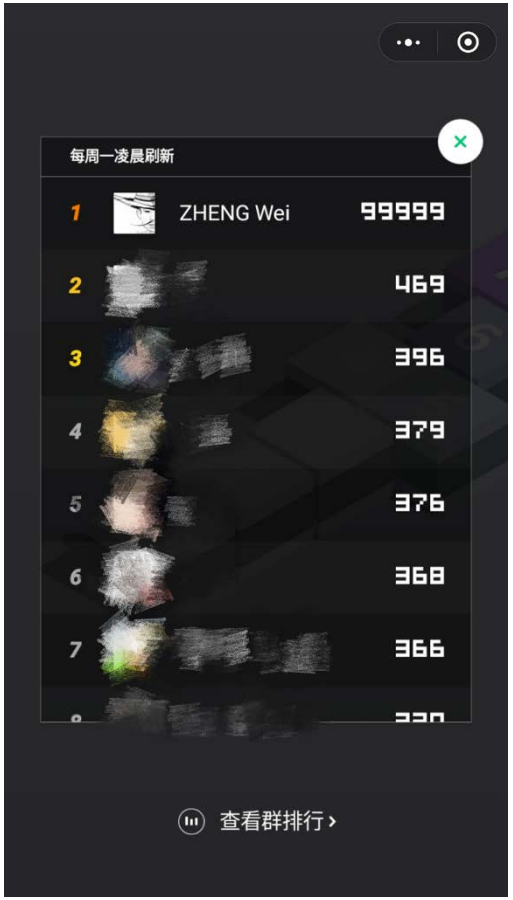
“教育就是当一个人把在学校所学全部忘光之后剩下的东西”



-- 爱因斯坦

编程与计算机科学

- 计算机科学代表了一种新的思考方式
- 编程是学习计算机科学的一种途径
- 学习编程不是终极目标
- 学编程的好处
  - ◆DIY程序应用
  - ◆对技术问题有深入理解
  - ◆高性价比的职业生涯



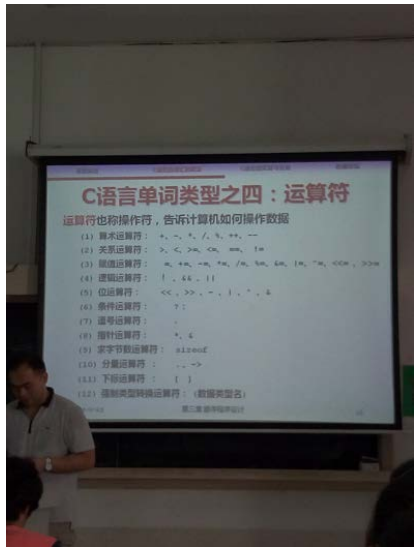
C语言该学但没有讲的内容

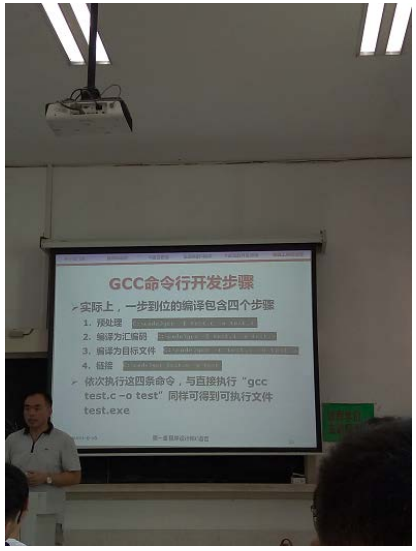
- 位运算符
  - 取反，与运算，或运算，移位操作
- 编译预处理
  - 关于#include和#define等命令的细节
- 更多的指针
  - 指向指针的指针，指针数组，指向函数的指针
- 多文件编程
- 还有很多……

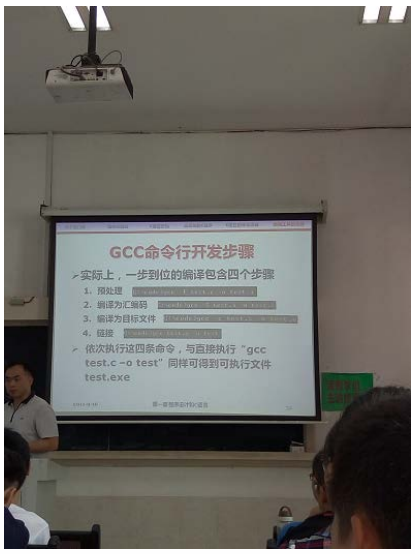
推荐后继阅读书籍

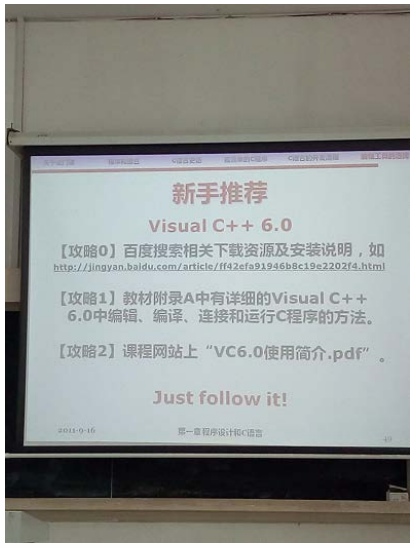
- C程序设计语言（第2版·新版）Brian W．Kernighan, Dennis M．Ritchie 徐宝文 李志 译 机械工业出版社
- C语言程序设计：现代方法（第二版），King 著，吕秀峰 译 人民邮电出版社
- C Primer Plus中文版（第五版）Stephen Prata 人民邮电出版社
- C专家编程 Peter Van Der Linden 徐波 译 人民邮电出版社
- C陷阱与缺陷 Andrew Koenig 高巍 人民邮电出版社
- C语言解惑 Alan R. Feuer 杨涛 译 人民邮电出版社
- 还有很多……

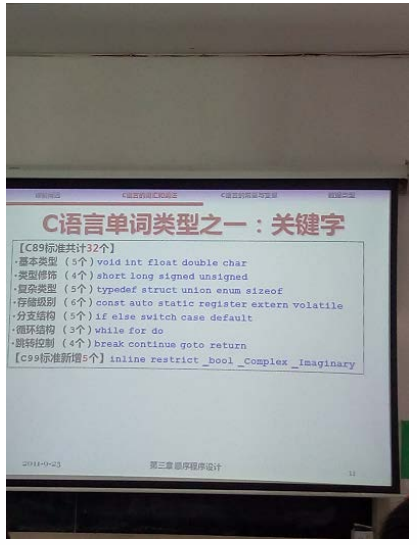
声明：拍照可以，请把我P掉











```
1. if((fp=fopen("test","r"))==NULL)
2. {
3.     printf("error!\n");
4.     exit(0);
5. }
6. ch=getchar();
7. while(ch!='#')
8. {
9.     fputc(ch,fp);
10.}
11.fclose(fp);
```

使用文件时忘记打开或打开方式与使用情况不匹配。

```
1. if((fp=fopen("test",“W"))==NULL)
2. {
3.     printf("error!\n");
4.     exit(0);
5. }
6. ch=getchar();
7. while(ch!='#')
8. {
9.     fputc(ch,fp);
10.}
11.fclose(fp);
```

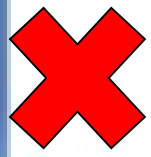
```
1. FILE fp;
2. if((fp=fopen("test","r"))
3.                         ==NULL)
4. {
5.     printf("cannot open
6.             this file!\n");
7.     exit(0);
8. }
```

误以为fopen返回的是FILE结构体变量。

```
1. FILE * fp;
2. if((fp=fopen("test","r"))
3.                         ==NULL)
4. {
5.     printf("cannot open
6.             this file!\n");
7.     exit(0);
8. }
```

```
1. if((fp=fopen("test","r"))
2.                          ==NULL)
3. {
4.     printf("cannot open
5.             this file!\n");
6.     exit(0);
7. }
```

//假设文件test和可执行程序并不在同一个目录

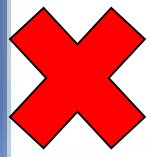


在打开文件时，指定的文件名找不到。

```
1. if((fp=fopen("d:\\test","r"))
2.                          ==NULL)
3. {
4.     printf("cannot open
5.             this file!\n");
6.     exit(0);
7. }
```

如果想打开的文件不在程序所在的目录下，应在指定文件名时指出文件路径

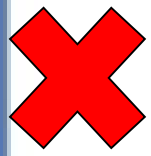
```
1. if((fp=fopen("test","w"))
2.                          ==NULL)
3. {
4.     printf("error!\n");
5.     exit(0);
6. }
7. fputs("hello\n", fp);
```



忘记关闭文件，可能会丢失数据。

```
1. if((fp=fopen("test","w"))
2.                          ==NULL)
3. {
4.     printf("error!\n");
5.     exit(0);
6. }
7. fputs("hello\n", fp);
8. fclose(fp);
```

```
1. int a[3]={1,3,5};
2. if((fp=fopen("test","w"))
3.                          ==NULL)
4. {
5.     printf("error!\n");
6.     exit(0);
7. }
8. fwrite(a[0],sizeof(int),
9.                          1,fp);
```



fread和fwrite函数的第一个参数是地址而不是变量。

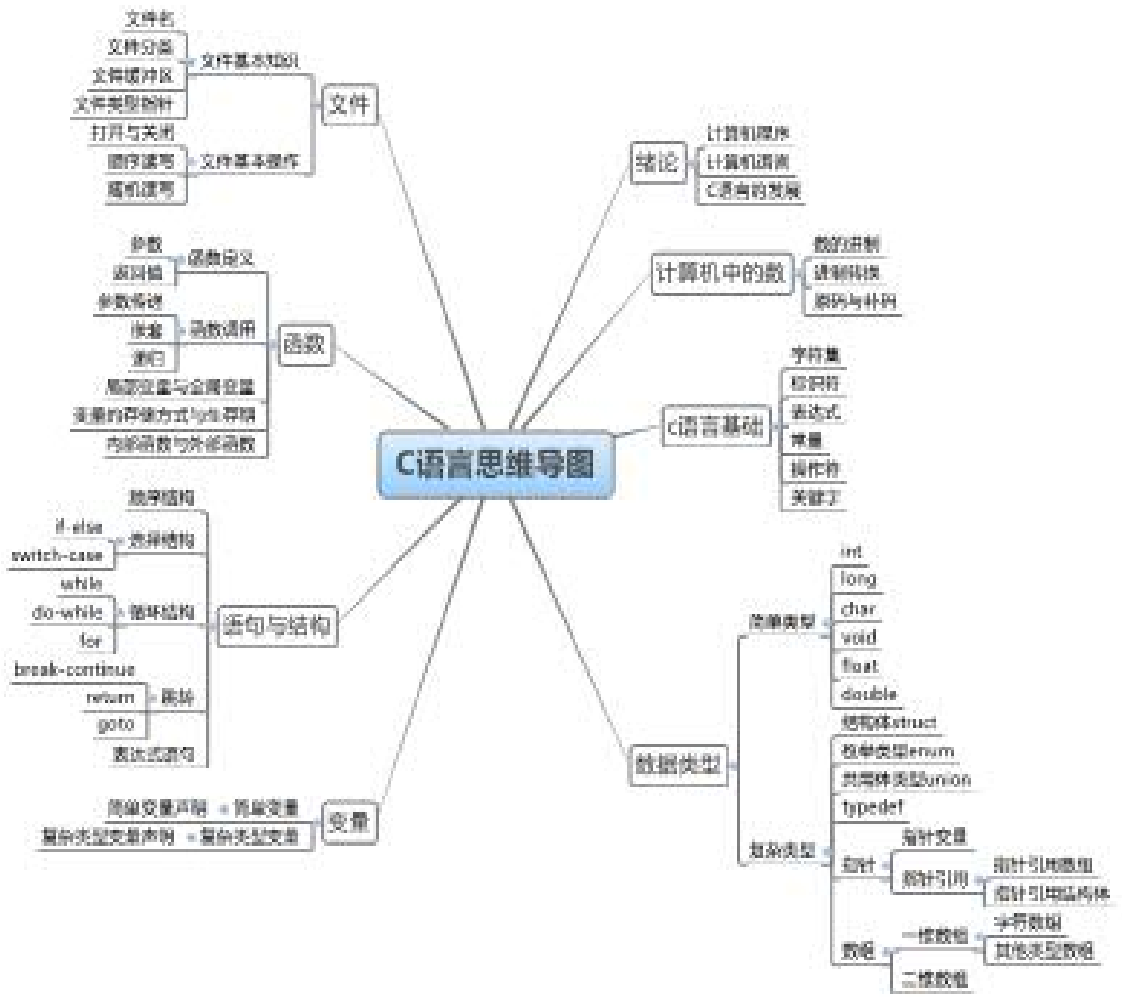
```
1. int a[3]={1,3,5};
2. if((fp=fopen("test","w"))
3.                          ==NULL)
4. {
5.     printf("error!\n");
6.     exit(0);
7. }
8. fwrite(&a[0],sizeof(int),
9.                          1,fp);
```

无尽模式

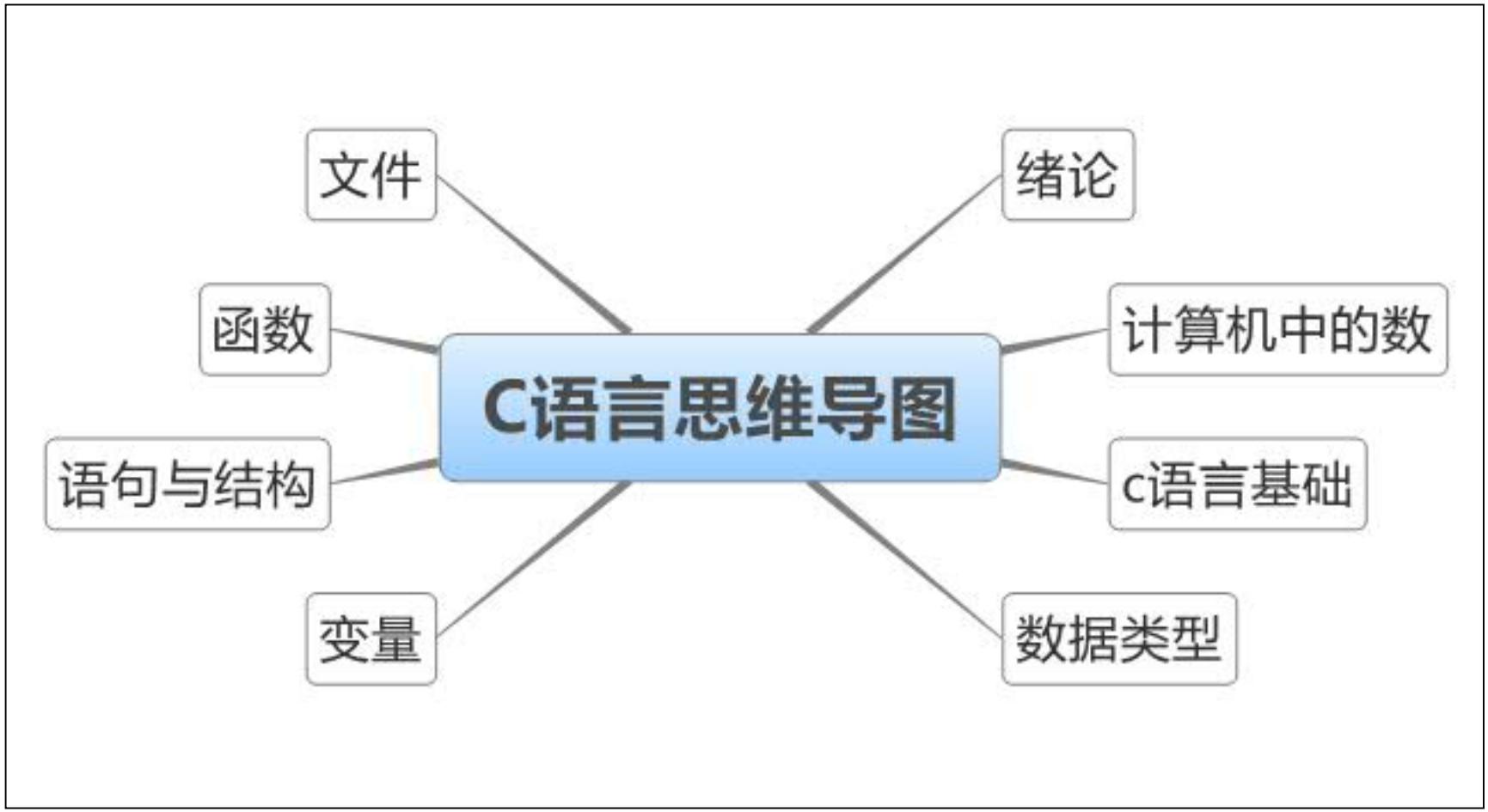
- C语言编程中容易犯的错误远不止于已经列举的这些，需要在实践中不断总结…

Good Luck!

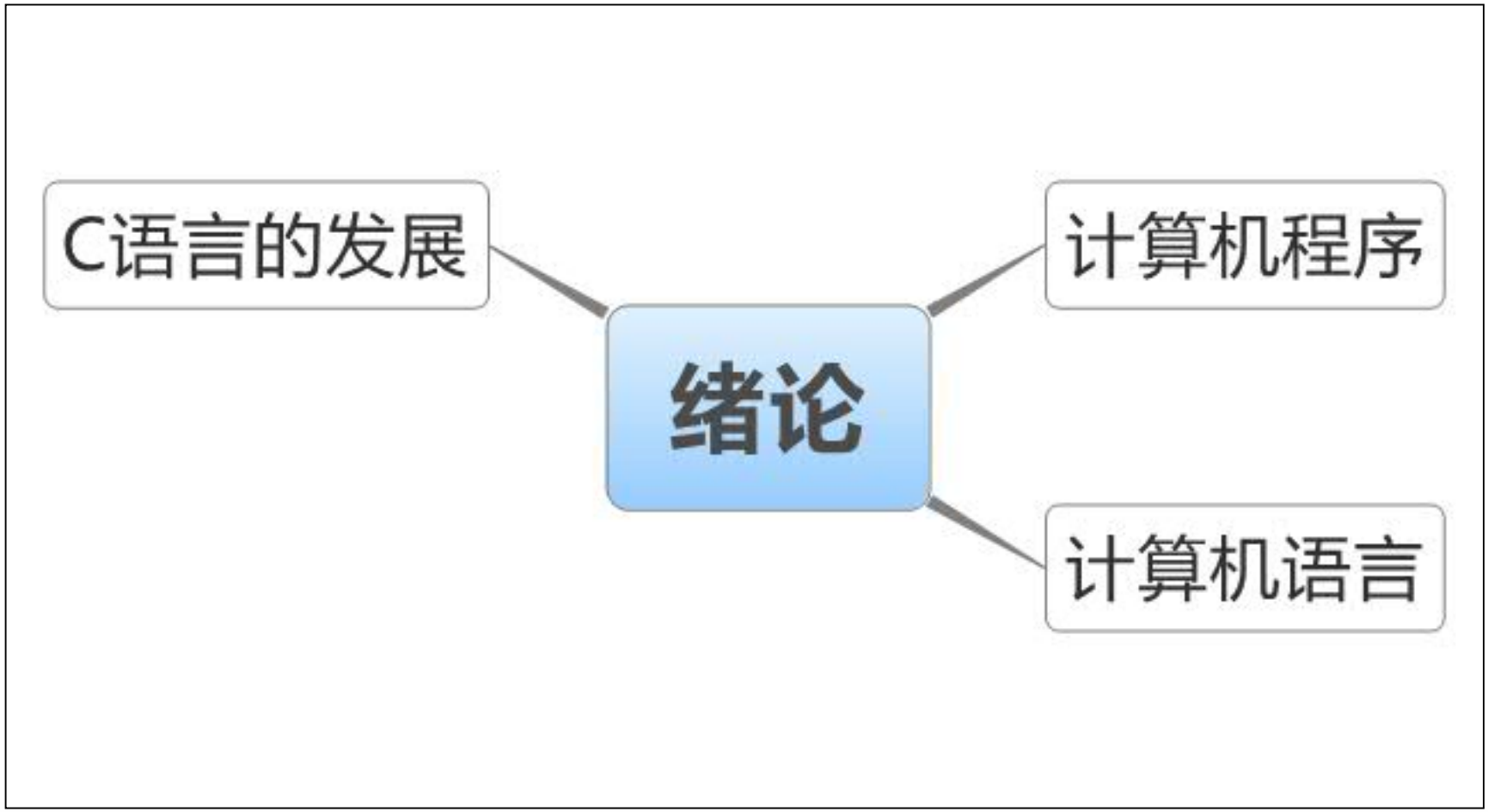
C语言思维导图


C语言思维导图
绪论
计算机中的数
C语言基础
数据类型
语句与结构
函数
变量
文件
int
long
char
void
float
double
typedef
if-else
switch-case
while
do-while
for
break-continue
return
goto

C语言思维导图



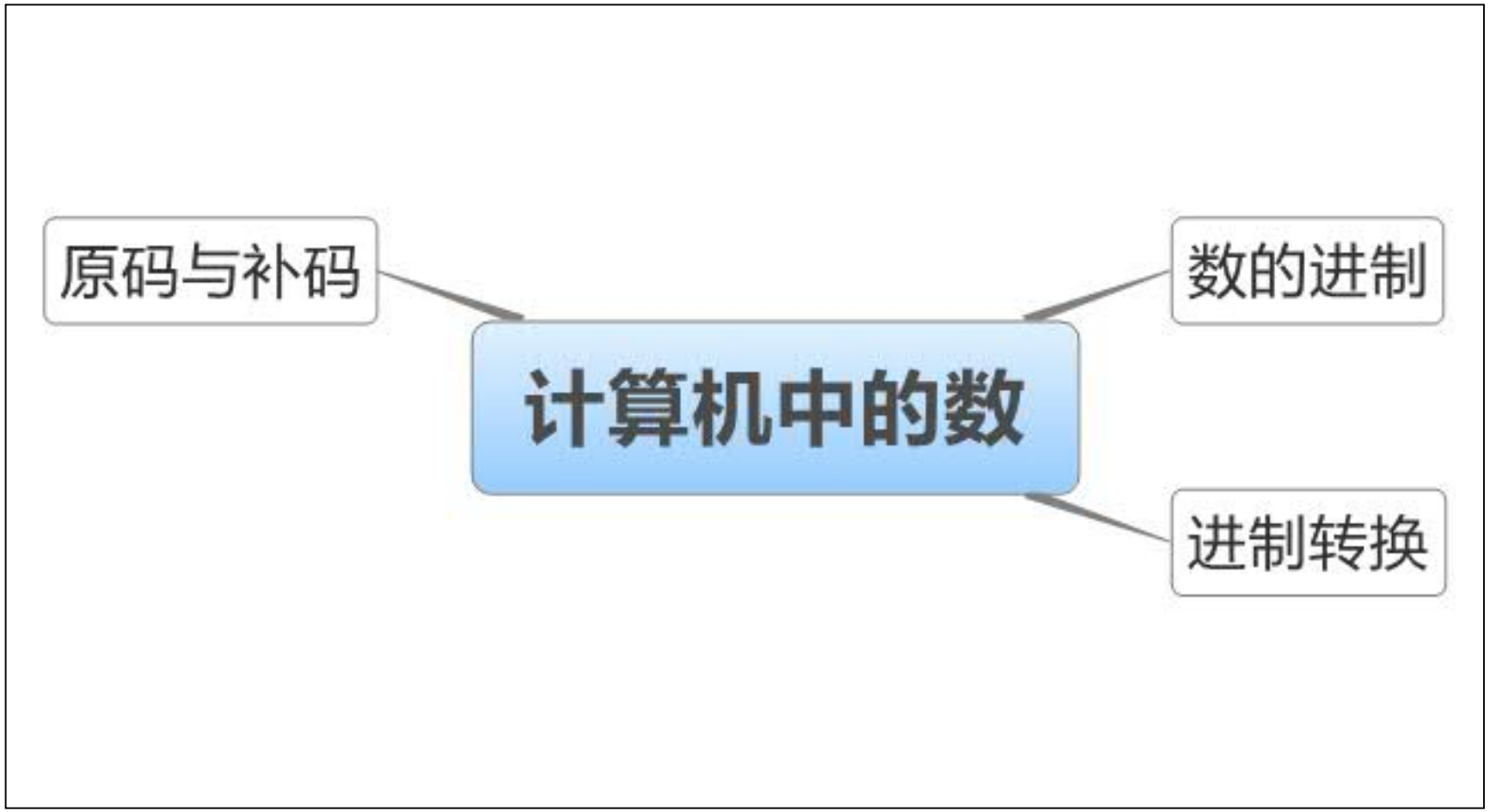
绪论



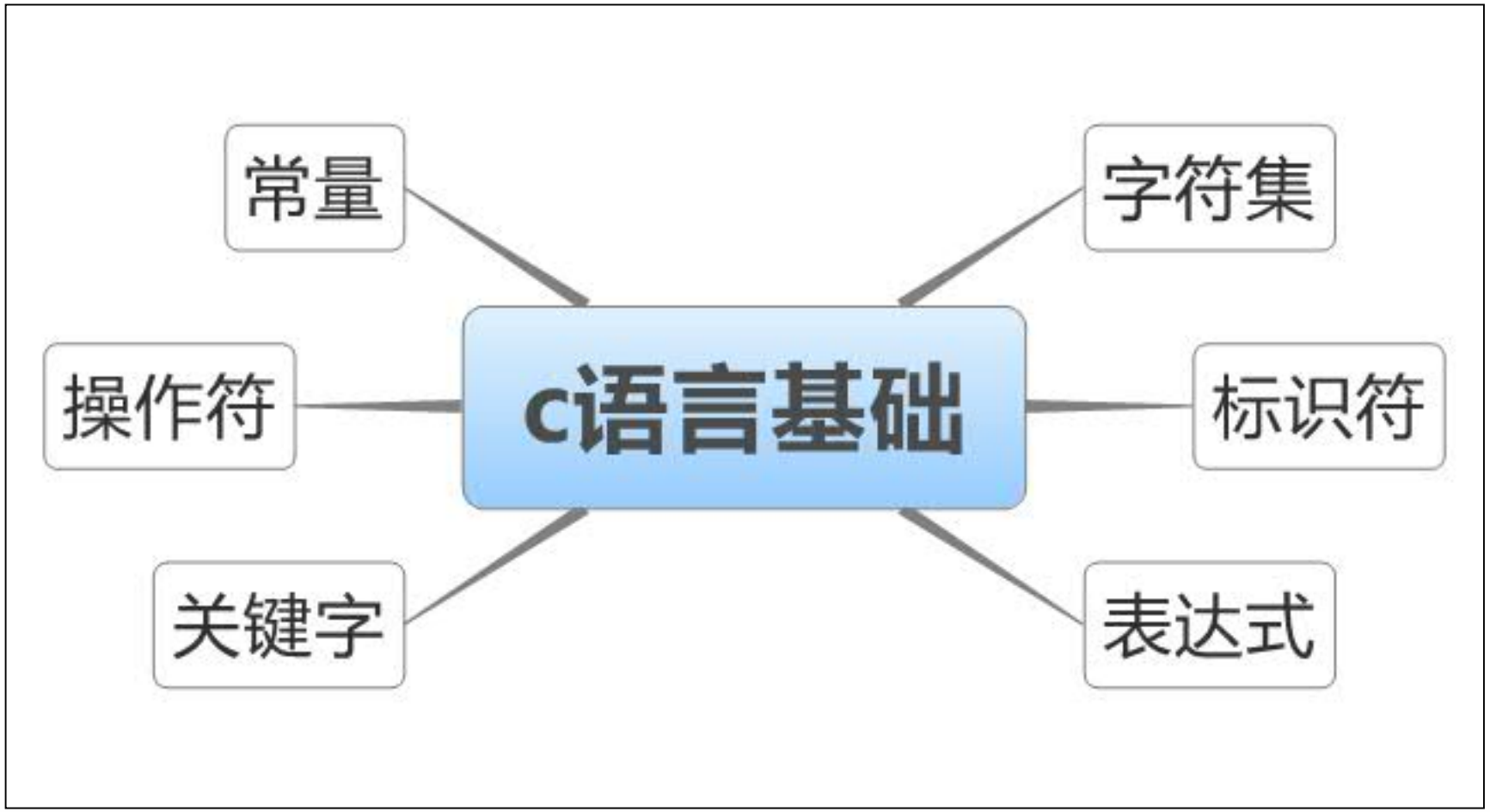
要点复习1

- 第2章：计算机中的数
  - 位（0，1），字节，补码，ASCII码
- 第3章：顺序结构程序设计
  - 数据表现形式：0,'0',"0"的区别？
  - 数据运算：整数除法的特点，类型转换
  - 运算结果溢出了会发生什么？
  - 表达式的值：赋值语句的值，++与--
  - scanf与printf：格式控制符，参数列表

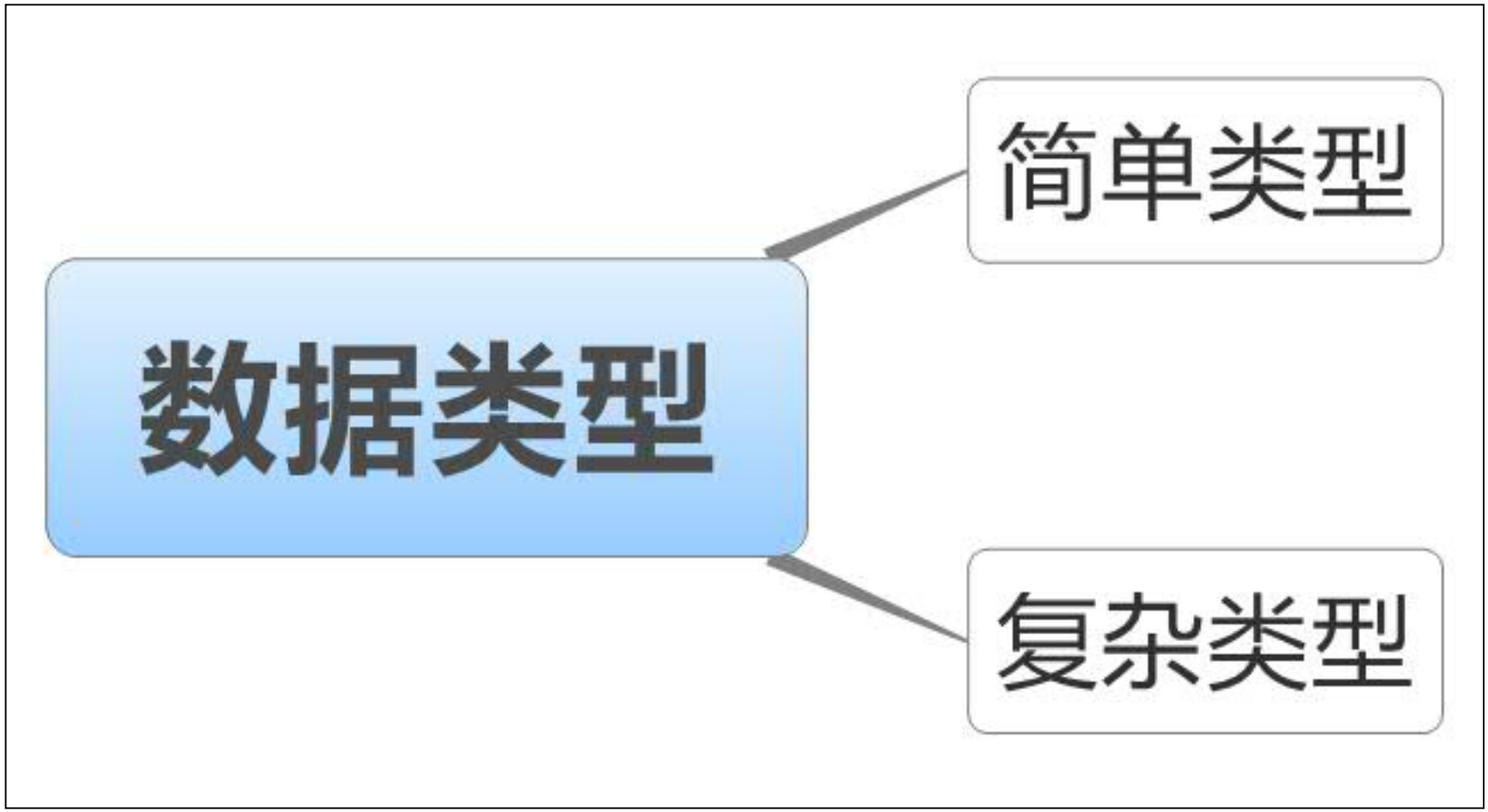
计算机中的数



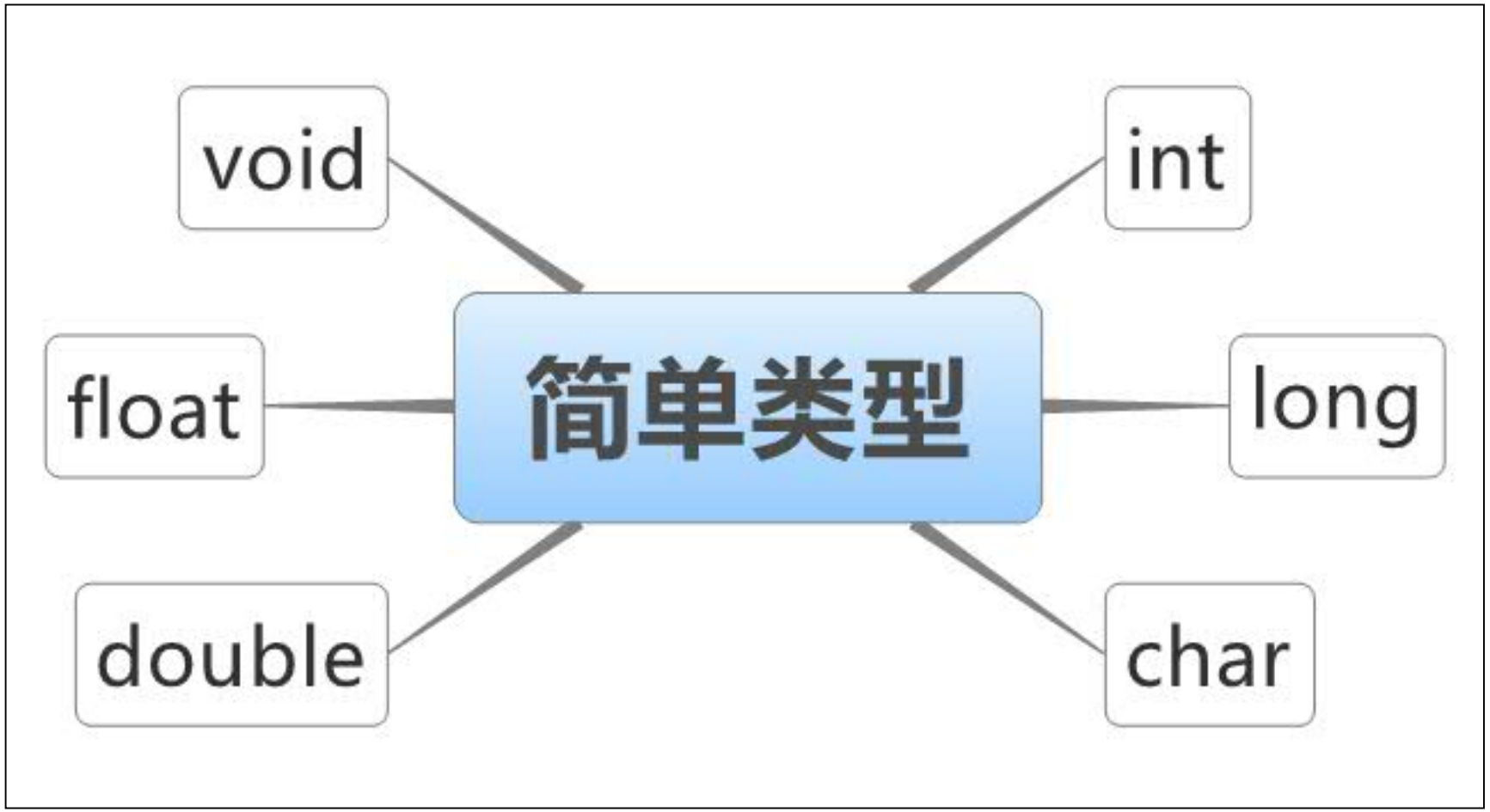
c语言基础



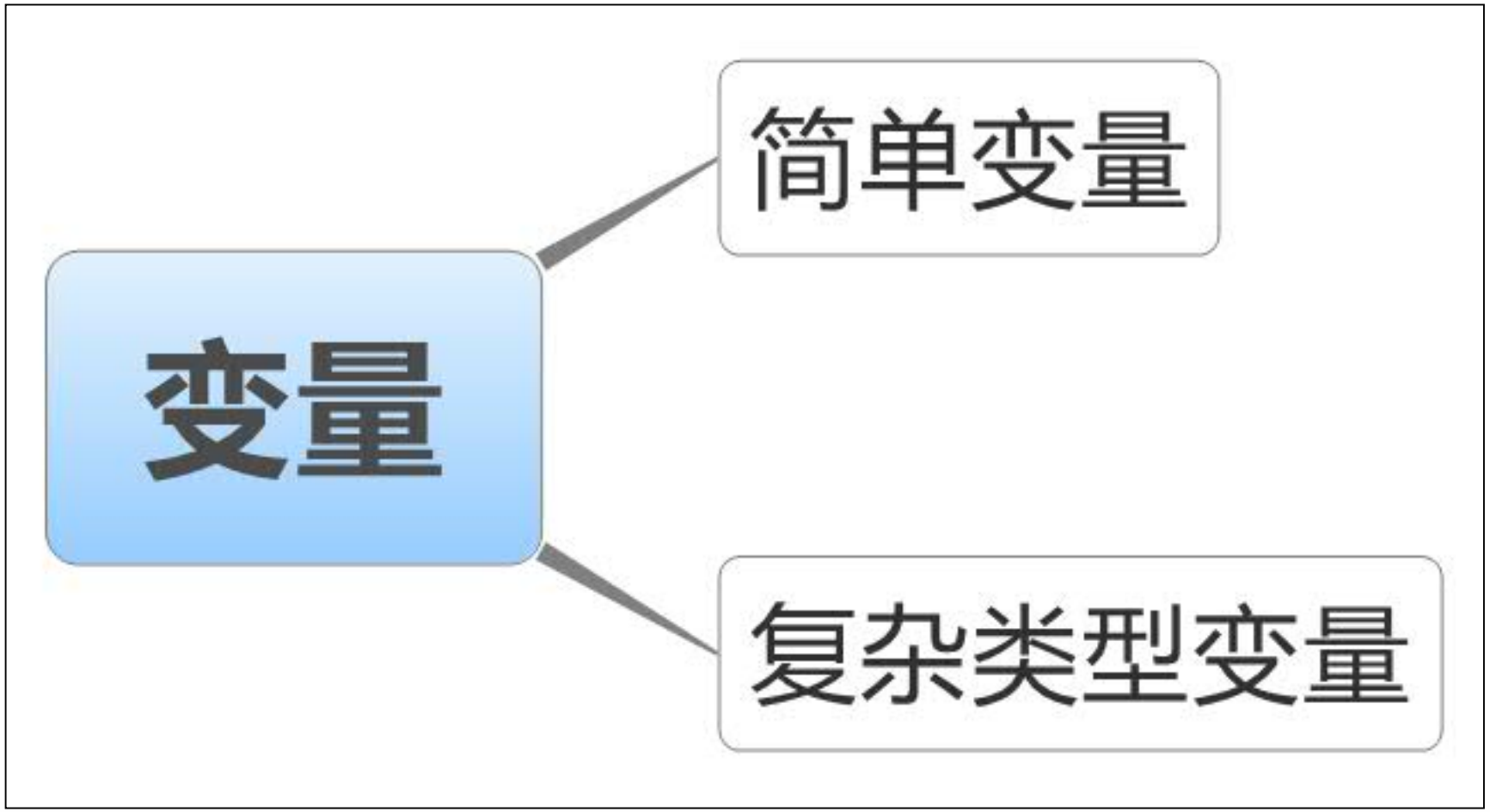
数据类型



简单类型



变量



简单变量

例题选讲1

- 下列代码的运行结果是什么？为什么？

```
#include <stdio.h>

int main()
{
    char x=0xFFFF;

    printf("%d\n", x--);

    return 0;
}
```

-1

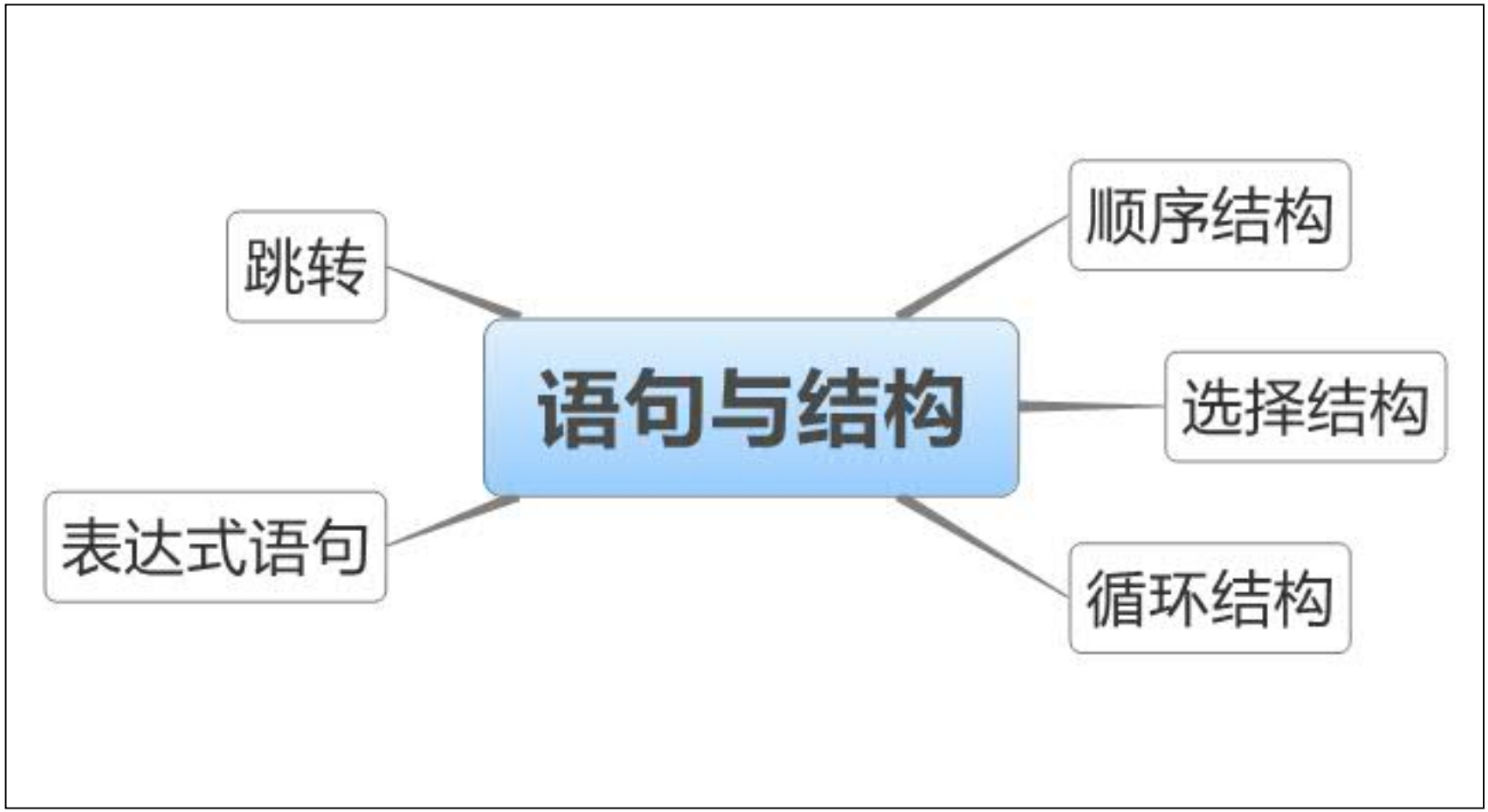
例题选讲2

- 下列代码的运行结果是什么？

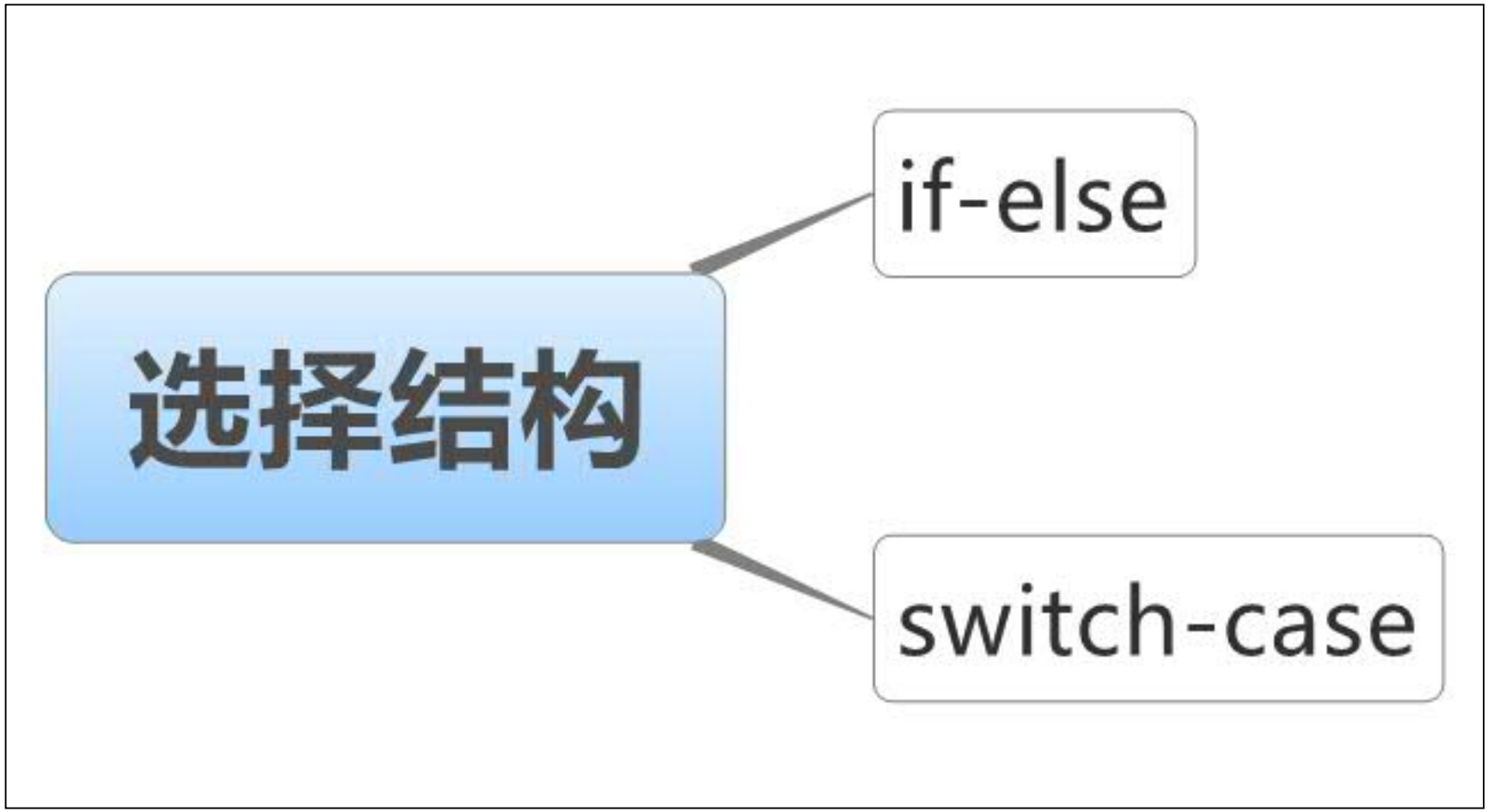
```
#include <stdio.h>
int main()
{
        int i=1, j=2, k=3;
        float m, f;
        f=i-k/j;
        printf("%f\n", f);
        m=j>1<k;
        printf("%f\n", m);
        f=k;
        printf("%f\n", f);
        f=i-f/j;
        printf("%f\n", f);
        return 0;
}
```

```
0.000000
1.000000
3.000000
-0.500000
```

语句与结构



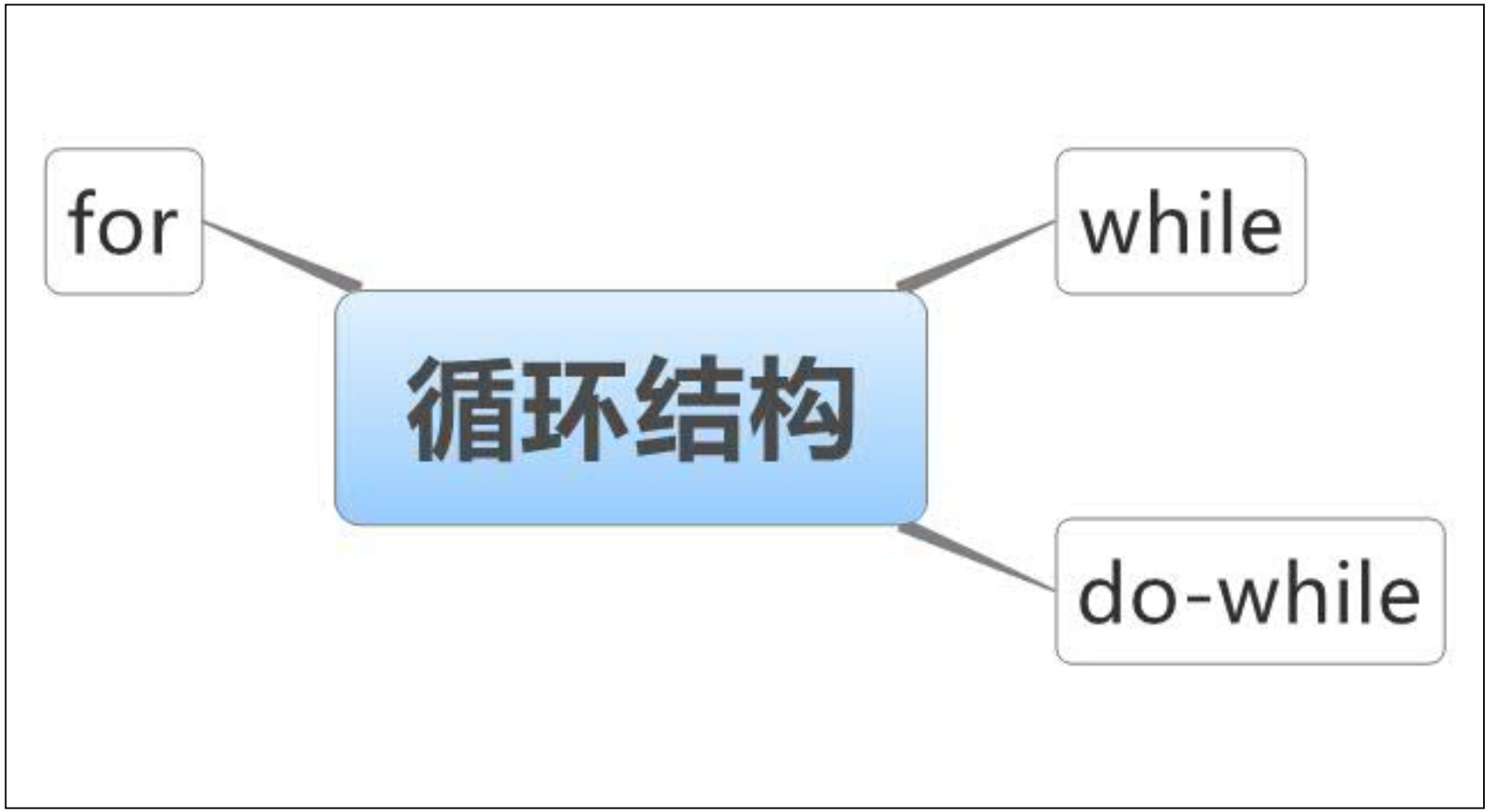
选择结构



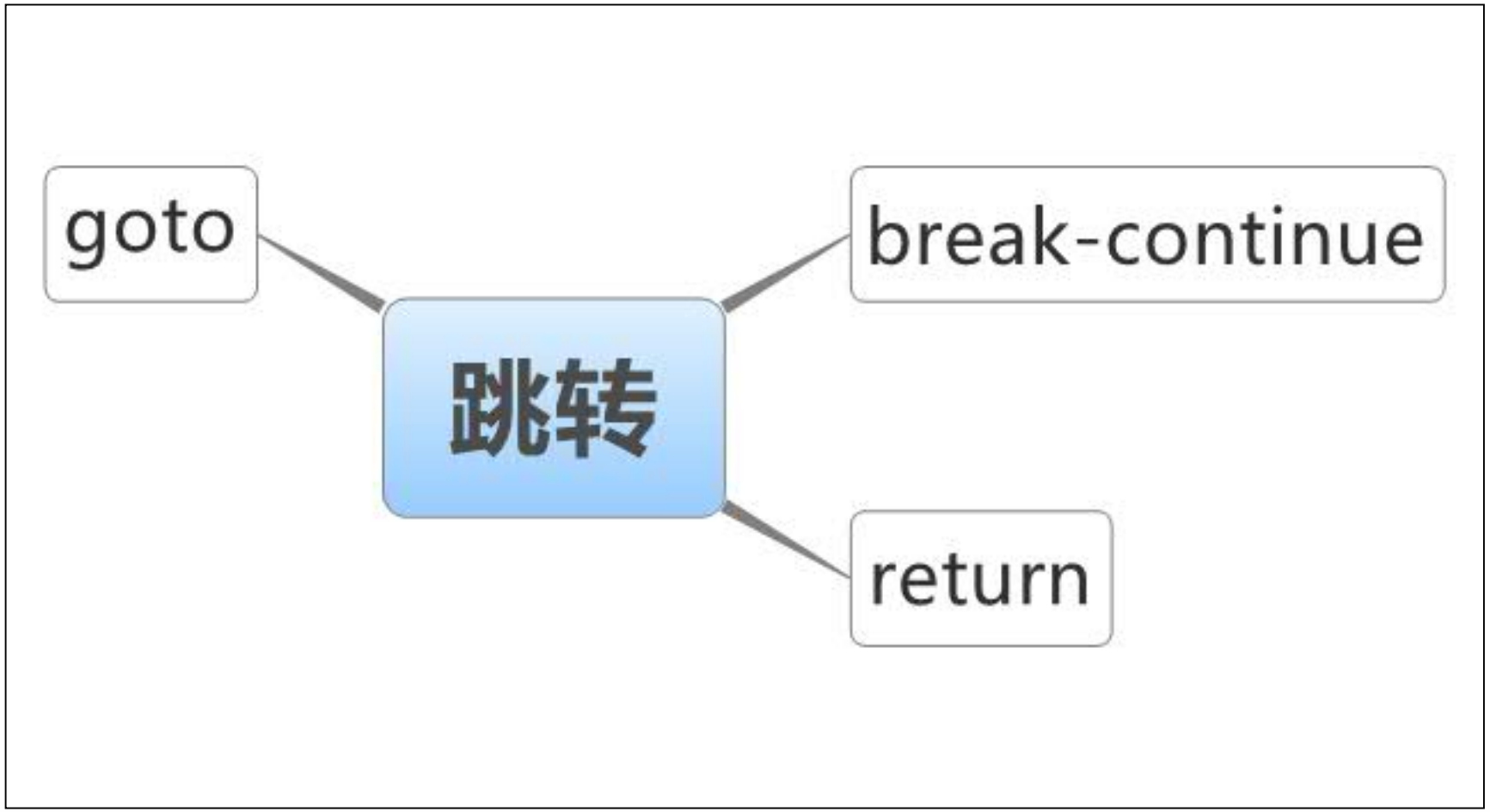
要点复习2

- 第4章：选择结构程序设计
  - 关系运算符：0＜a＜2为什么错了？
  - if 右边括号内的表达式如何理解？
  - 逻辑运算符的短路运算现象？0&&...,1||...
  - 选择结构的嵌套：if...else...如何匹配？
  - switch语句：表达式的类型应该为整型，case后面应该是常量，case分支中带不带break的区别，default的作用

循环结构



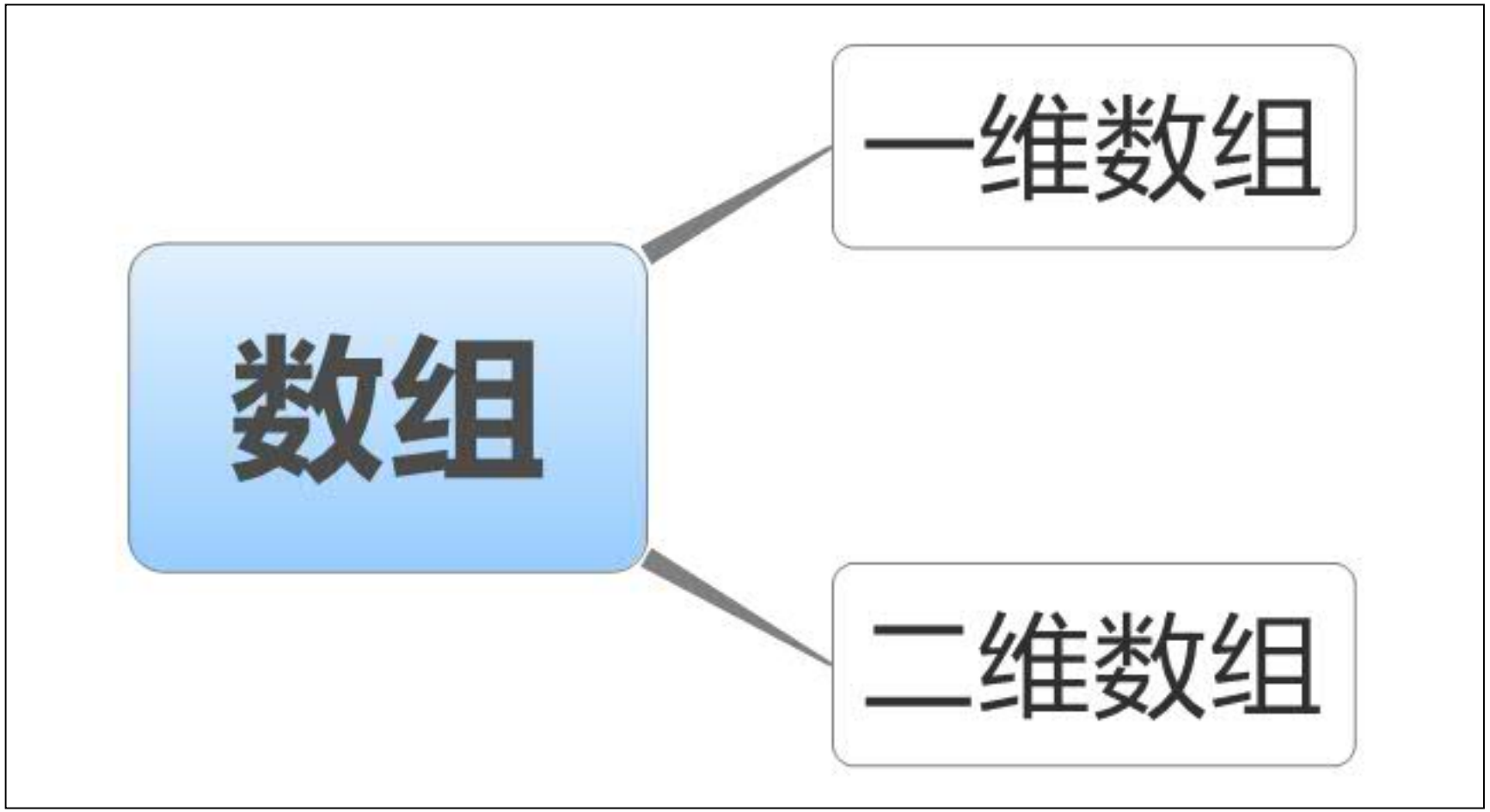
跳转



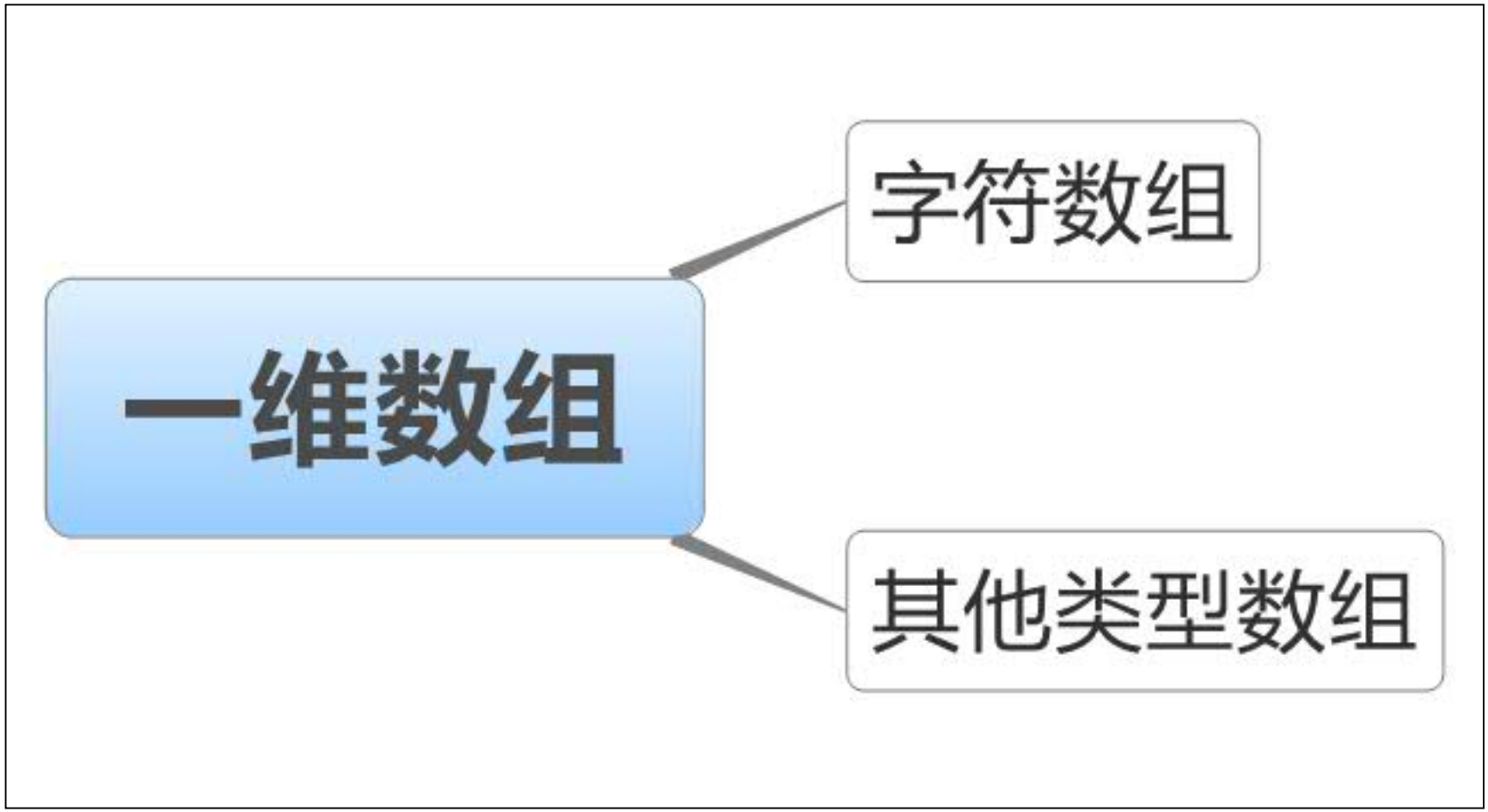
要点复习3

- 第5章：循环结构程序设计
  - While和if的区别？
  - While和do...while的区别？
  - For语句的优点是什么？
  - 小心循环的边界
  - 循环的嵌套
  - Break与continue的作用
  - 循环结构与选择结构的结合

数组



一维数组



要点复习4

- 第6章：利用数组处理批量数据
  - 数组的下标从0开始
  - 长度为n的数组a，最后一个元素是a[n-1]
  - 数组的初始化与引用
  - 数组名不是变量，不能赋值
  - 字符数组及其处理函数
  - 字符串的结束标记'\0'

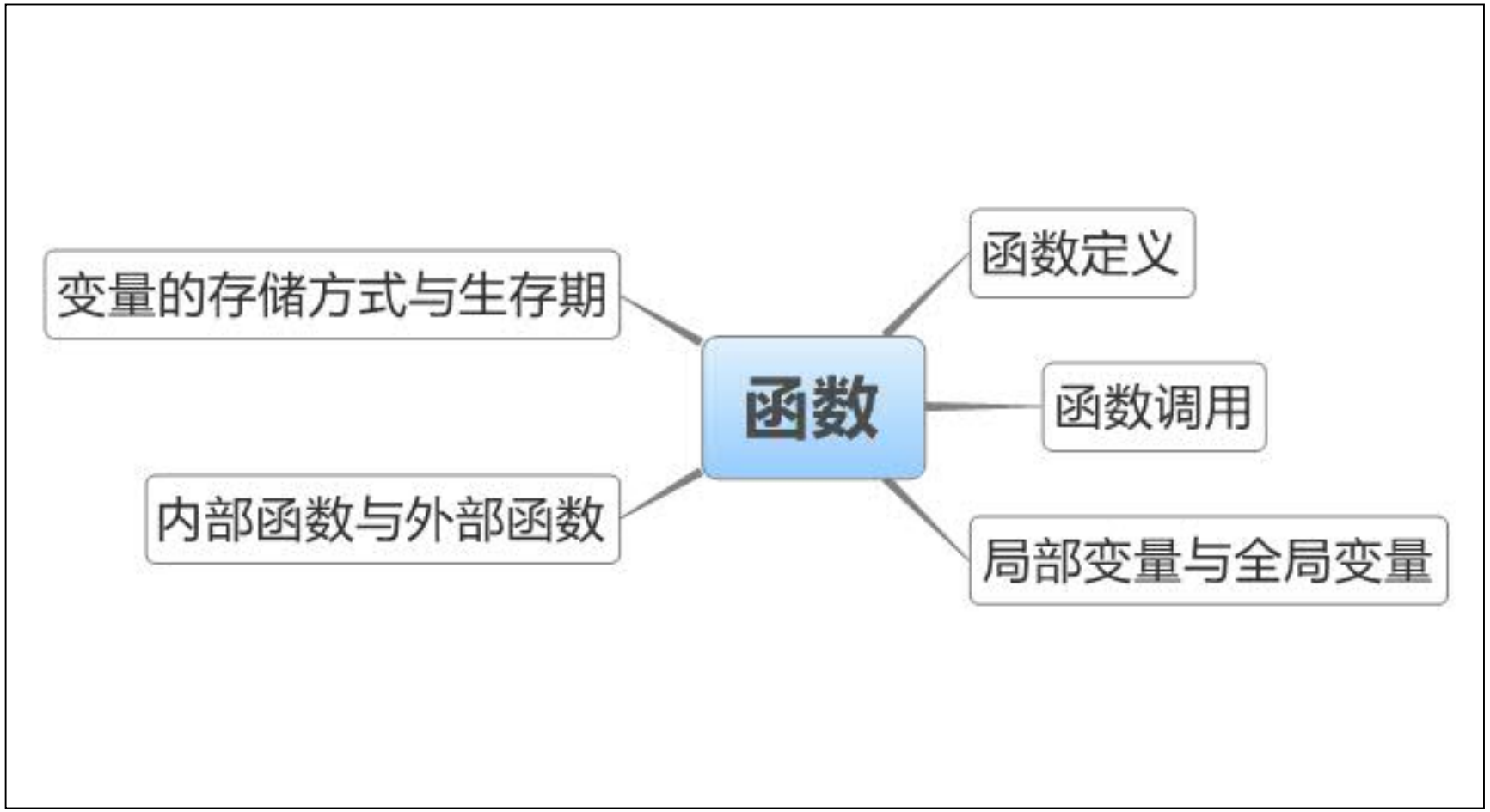
例题选讲3

- 下列代码的运行结果是什么？

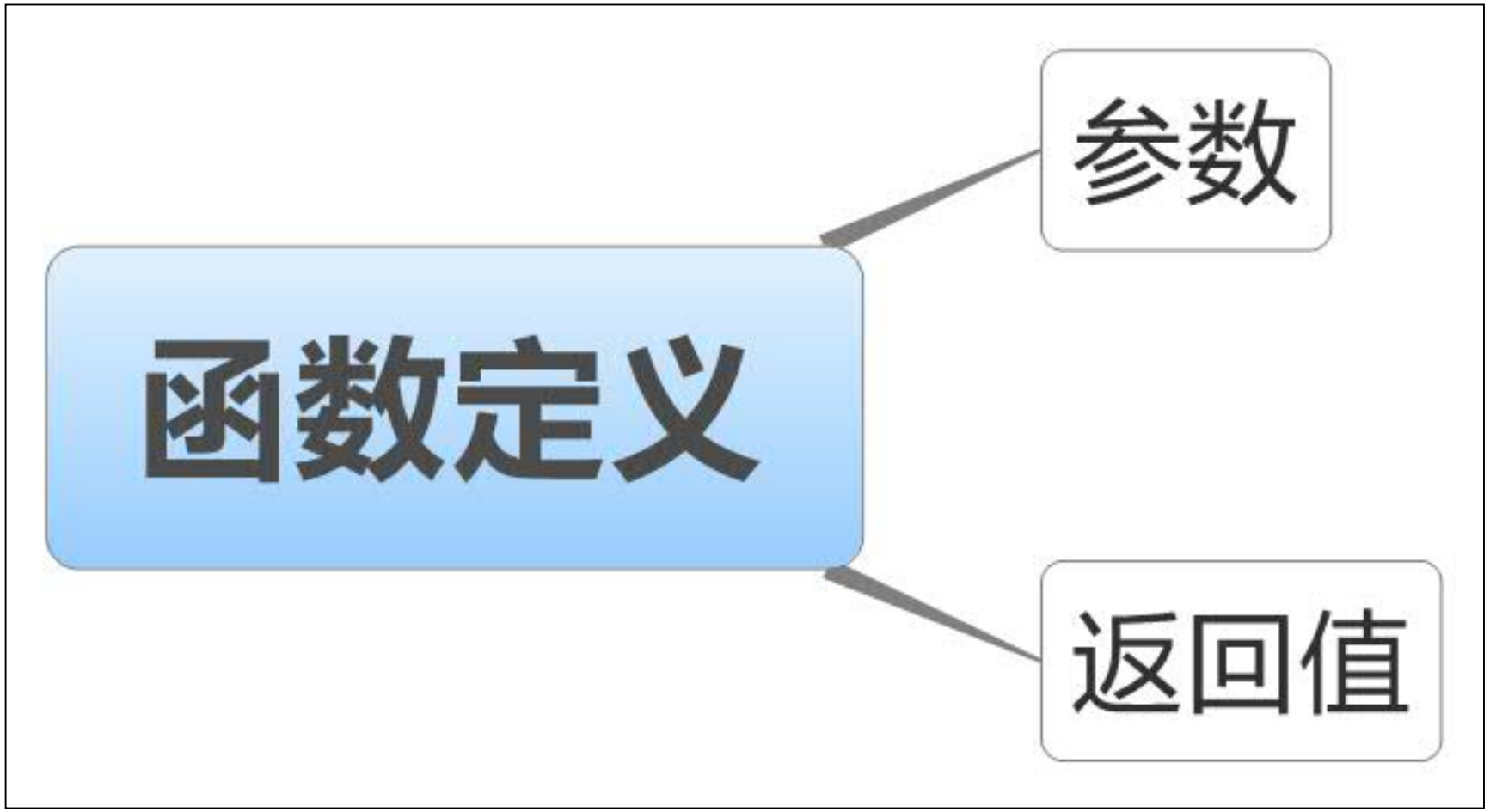
```
int a[][3]={9,7,5,3,1,2,4,6,8};
int i,j,s1=1,s2=0;
for(i=0;i<3;i++) {
  for(j=0;j<3;j++) {
     if(i+j==2) {
      s2+=a[i][j];
      continue;
     }
     if(i!=j) {
       s1*=a[i][j];
     }
  }
}
printf("%d,%d",s1,s2);
```

252,10

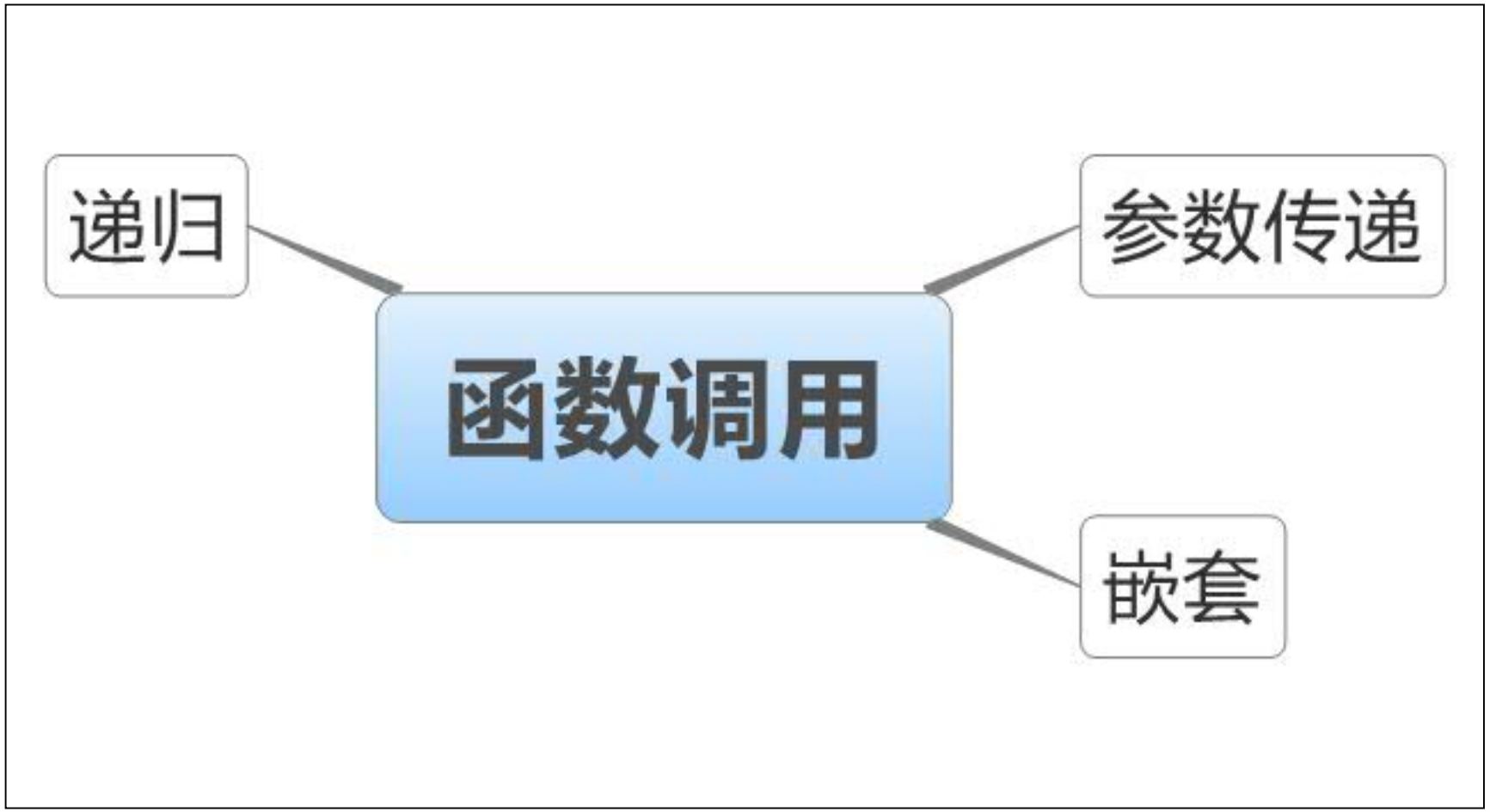
函数



函数定义



函数调用



例题选讲4

- 下列代码的运行结果是什么？

```
void f(char ch, int i)
{
        i++;
        if (ch>'A')
        {
                f(ch-i,i);
        }
        putchar(ch);
}
int main()
{
        f('G',0);
        return 0;
}
```

ADFG

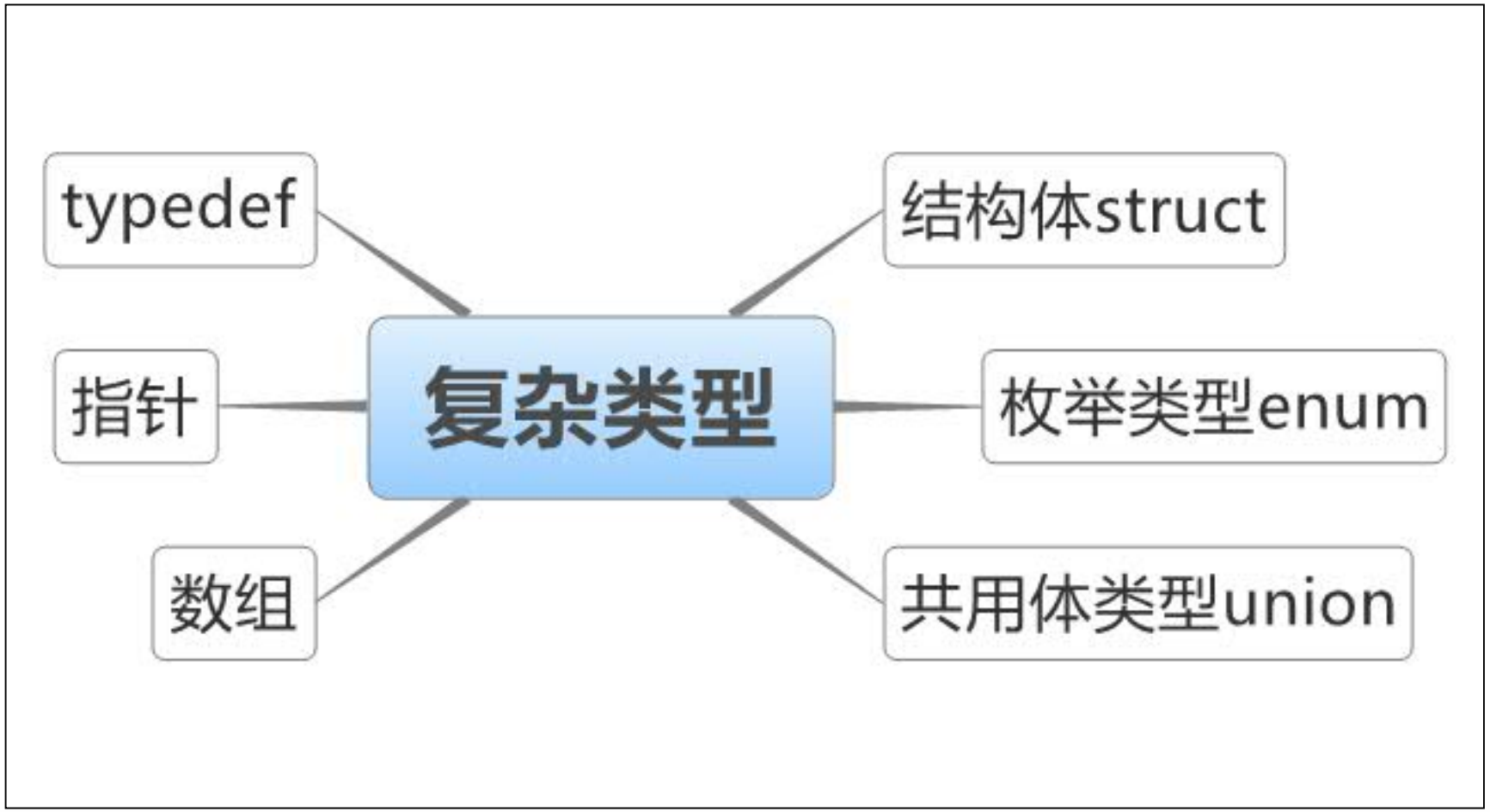
例题选讲5

- 下列代码的运行结果是什么？

```
#include <stdio.h>
static int i,j; int k = 0;
int fun1()
{
    static int i=0; i++; return i;
}
void fun2()
{
    j=0; j++;
}
int main()
{
    for (k=0; k<10; k++) { i=fun1(); fun2(); }
    printf("%d,%d", i, j);
    return 0;
}
```

10,1

复杂类型



复杂类型变量

例题选讲6

- 下列代码的运行结果是什么？

```
#include <stdio.h>

int main()
{
    struct cmplx{int x; int y;} cnum[2]={1,3,2,7};

    printf("%6.2f\n",
              (float)(cnum[0].y/cnum[1].x*cnum[1].y));
    return 0;
}
```

```
  7.00
```

例题选讲7

- 下列代码的运行结果是什么？

```
#include <stdio.h>

union myun
{
    struct { int  x,y,z; }u;
    int k;
}a;

int main()
{
    a.u.x=4; a.u.y=5; a.u.z=6; a.k=0;
    printf("%d\n", a.u.x);

    return 0;
}
```

0

例题选讲8

- 下列代码的运行结果是什么？

```
#include <stdio.h>

union pw {int i; char ch[2]; } a;

int main()
{
    a.ch[0] = 2;
    a.ch[1] = 1;
    a.ch[2] = 0;
    a.ch[3] = 0;

    printf("%d\n", a.i);

    return 0;
}
```

258

例题选讲9

- 下列代码的运行结果是什么？

```
#include <stdio.h>

int main()
{
    union {int a; long b; unsigned char c; } m;

    m.b = 0x12345678;

    printf("%d\n", m.c);

    return 0;
}
```

120

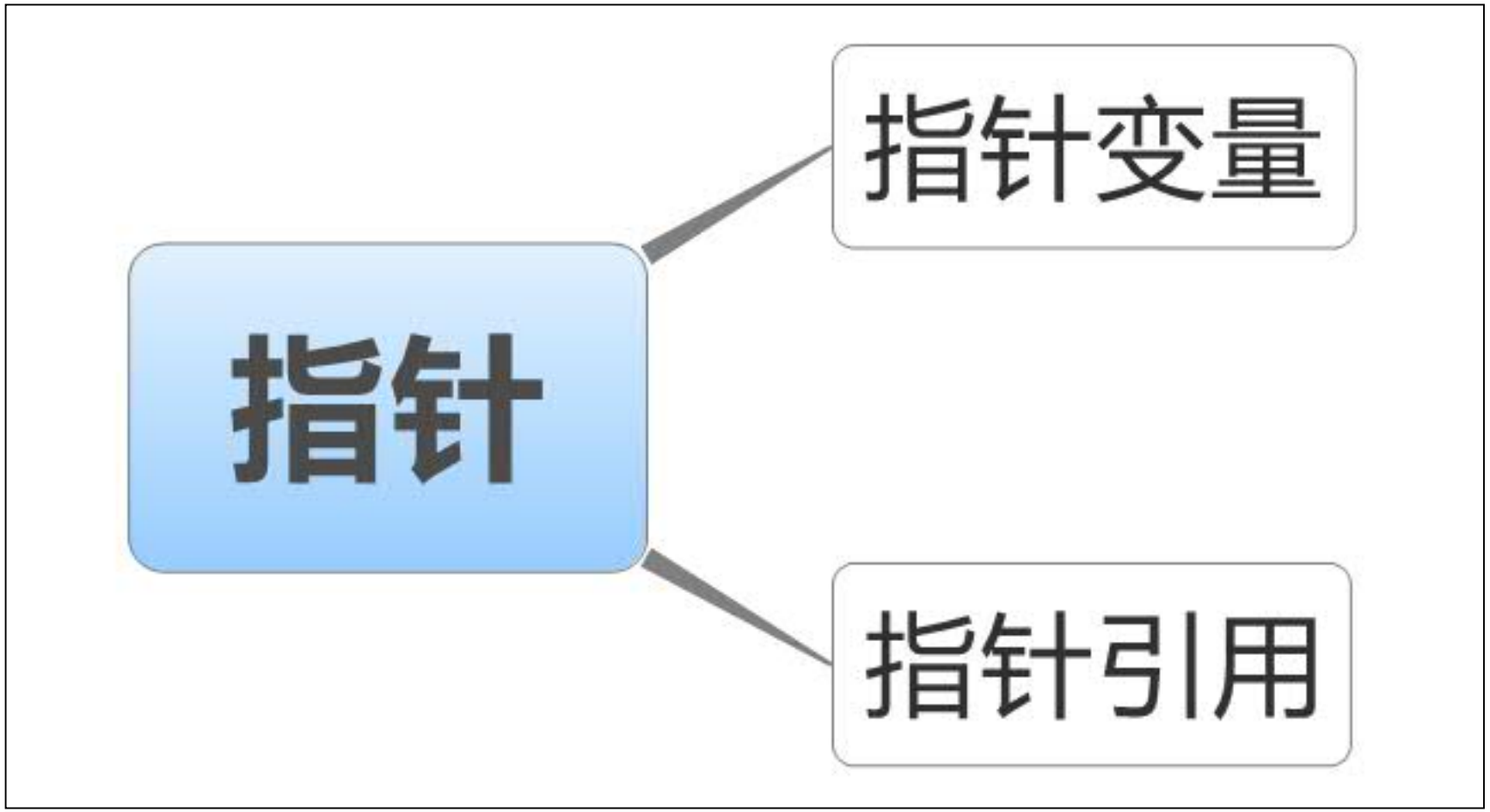
例题选讲10

- 下列代码的运行结果是什么？

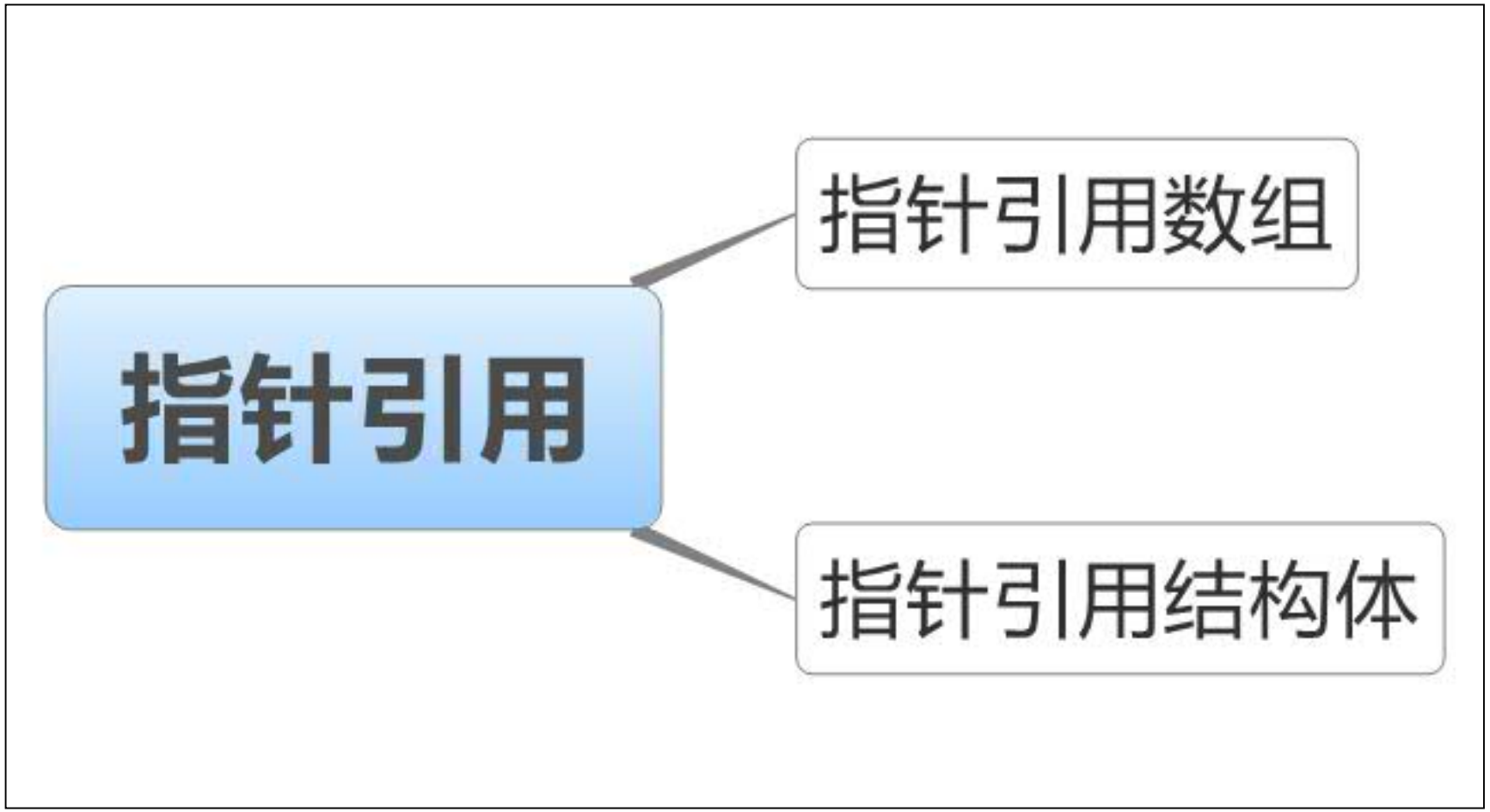
```
#include <stdio.h>

typedef union
{
    long a[2];
    int b[4];
    char c[8];
} TY;

int main()
{
    TY our;
    printf("%d\n", sizeof(our));

    return 0;
}
```

16

指针



指针引用



例题选讲11

- 下列代码的运行结果是什么？

```
void f(char *s)
{
    if (*s)
    {
        f(s+1);
        putchar(*s);
        f(s+1);
    }
}
int main()
{
    char *str="abc";
    f(str);
    return 0;
}
```

cbcacbc

例题选讲12

- 下列代码的运行结果是什么？

```
#include <stdio.h>

int main()
{
    int a[3][2] = {(0,1),(2,3),(4,5)};
    int *p;

    p = a[0];

    printf("%d", p[0]);

    return 0;
}
```

1

例题选讲13

- 设数组a的首地址为200，以下程序输出是？

```
void main()
{
    short a[3][3]={1,2,3,4,5,6,7,8,9};
    printf("%d %d\n", a, *a);
    printf("%d %d\n", a[1], a+2);
    printf("%d %d\n", &a[0], &a[1][0]);
    printf("%d %d\n", a[1][0], *(a+1)+2);
}
```

200, 200
206, 212
200, 206
4, 210

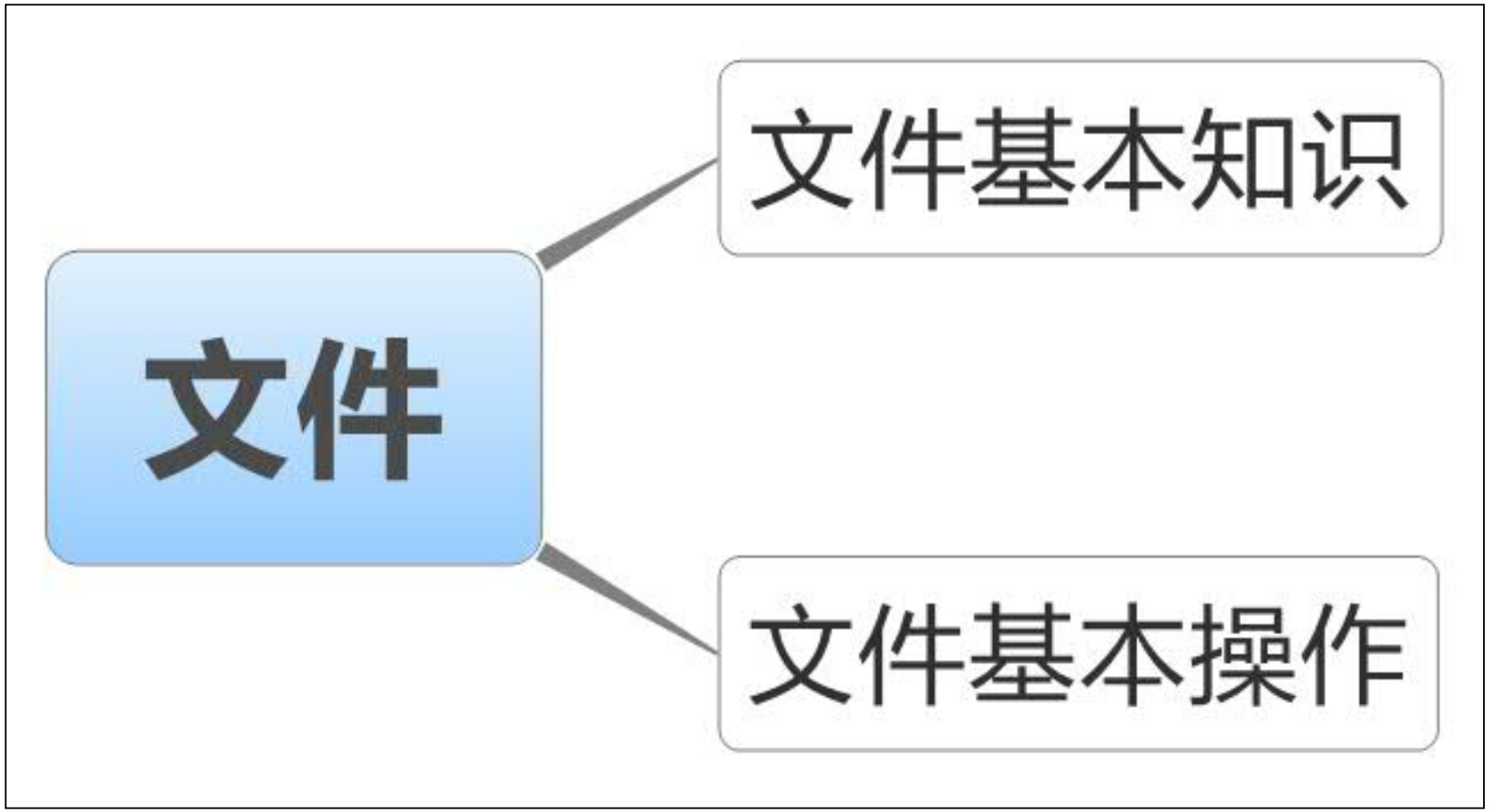
例题选讲14

- 下列代码的运行结果是什么？为什么？

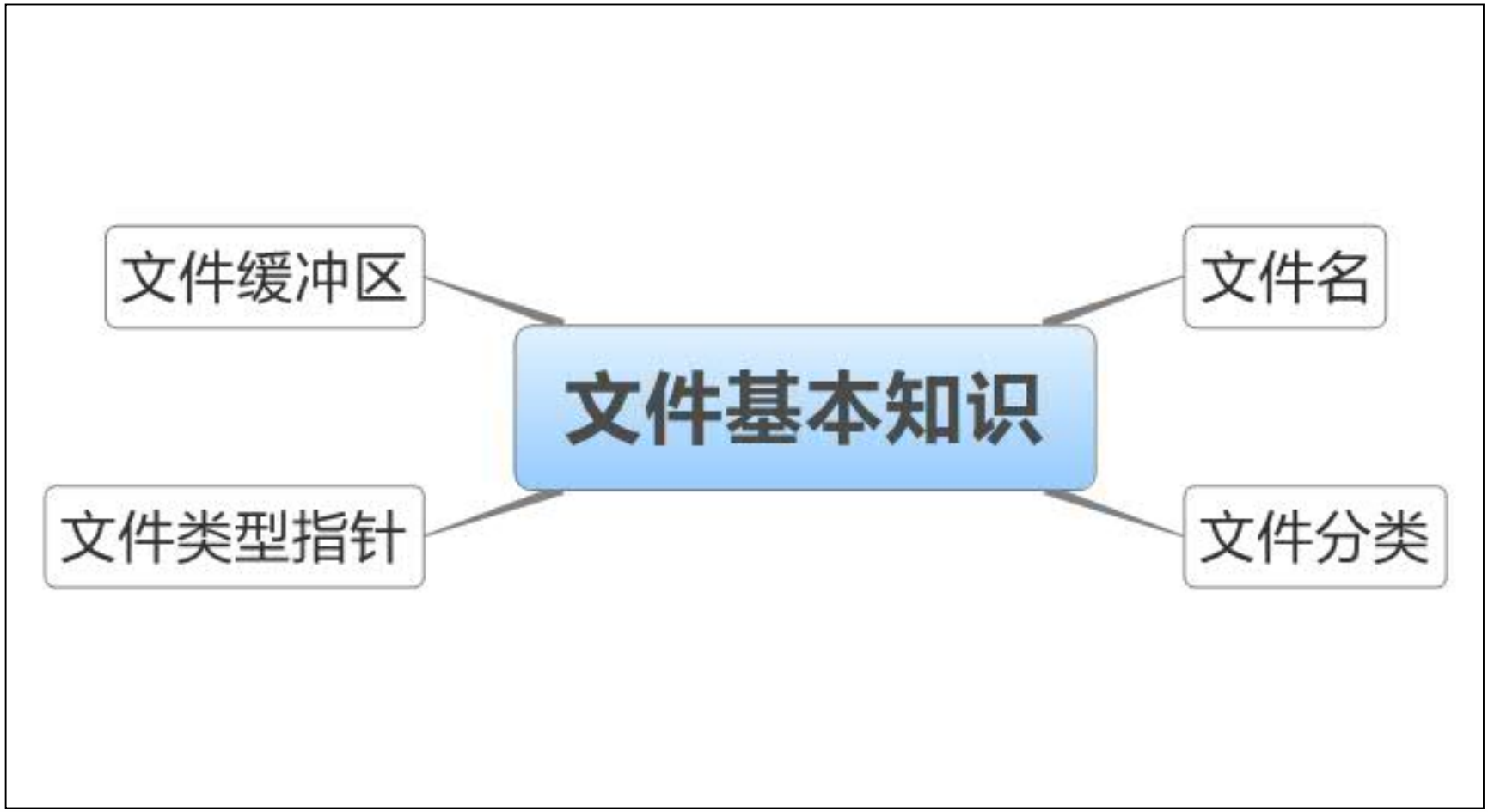
```
#include <stdio.h>

int main()
{
    double *p = NULL;
    int a[100];
    printf("%d, %d\n", sizeof(p), sizeof(*p));
    printf("%d, %d\n", sizeof(a), sizeof(a[100]));
    printf("%d, %d\n", sizeof(&a), sizeof(&a[0]));

    return 0;
}
```

```
4, 8
400, 4
4, 4
```

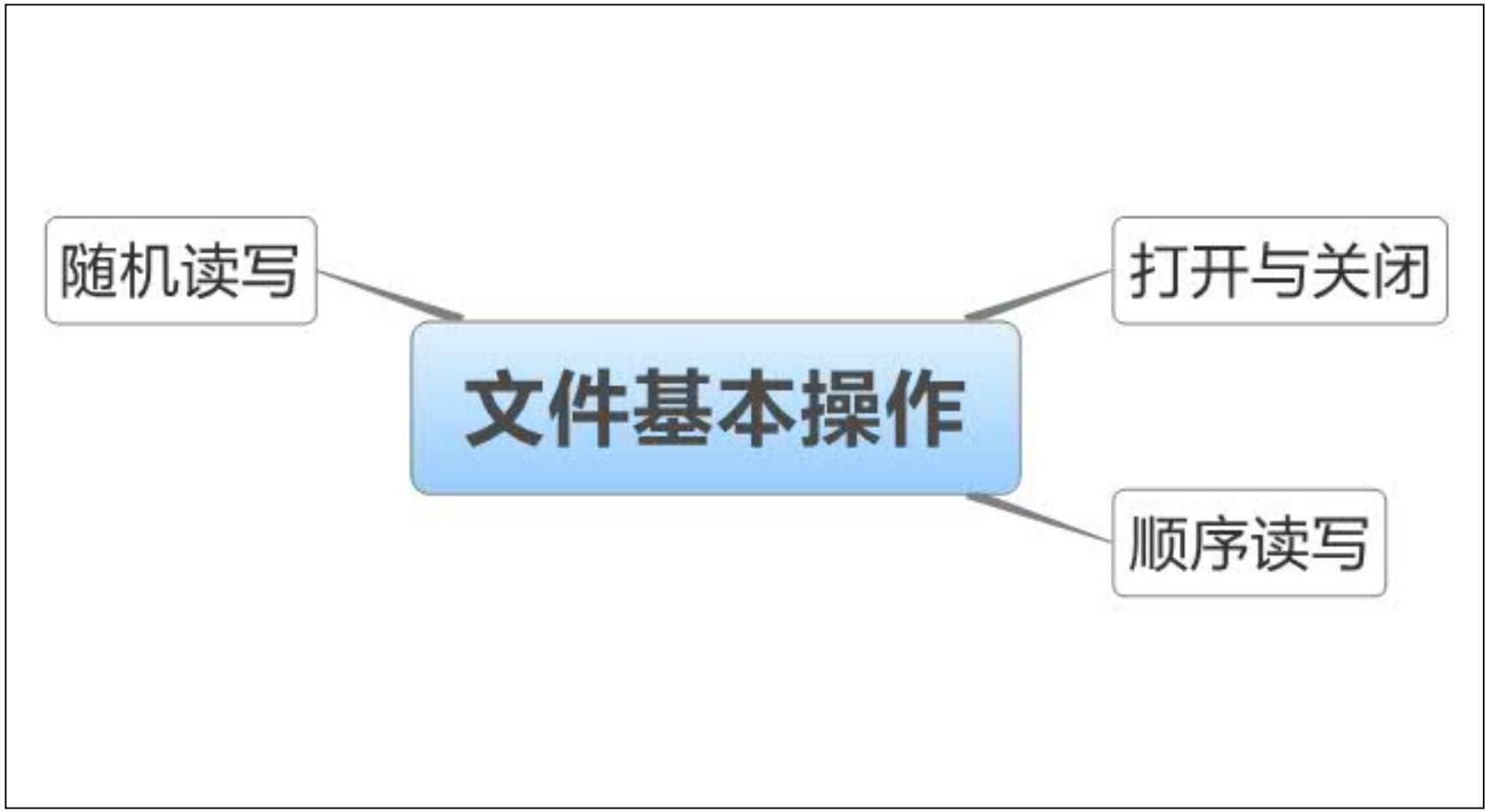
文件



文件基本知识



文件基本操作



例题选讲15

- 假定当前路径下已有文件a1.txt，其内容如下：
  ABc#dEF下列代码的运行结果是什么？

```
void f(FILE *fp)
{
    char c;
    while((c=fgetc(fp))!='#')
    {
        printf("%d ", c>='a'?c-'a'+10:c-'A'+10);
    }
}
int main()
{
    FILE *fp;
    fp=fopen("a1.txt","r");
    f(fp);
    fclose(fp);
    return 0;
}
```

```
10 11 12
```

例题选讲16

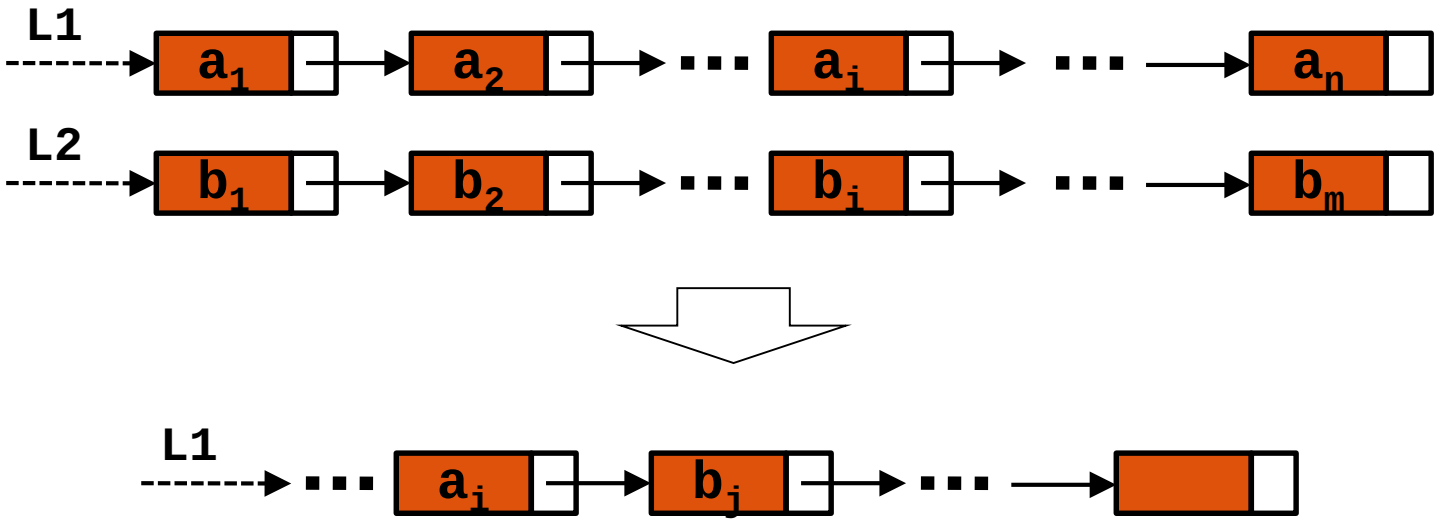
- 下列代码的运行结果是什么？

```
#include <stdio.h>

int main()
{
    FILE *fp; char c; int i,j;

    fp=fopen("a.tmp","w");
    fputs("3.14", fp);
    fputs("2.87", fp);
    fclose(fp);
    fp=fopen("a.tmp","r");
    fscanf(fp,"%d%c%d",&i, &c, &j);
    printf("%d,%c,%d\n",i,c,j);
    fclose(fp);
    return 0;
}
```

```
3,.,142
```

不带头结点的有序链表的合并

- 如图所示，链表L1和L2的结点数据顺序递增，写一个函数把L2合并到L1中，且保持结点数据的顺序递增。

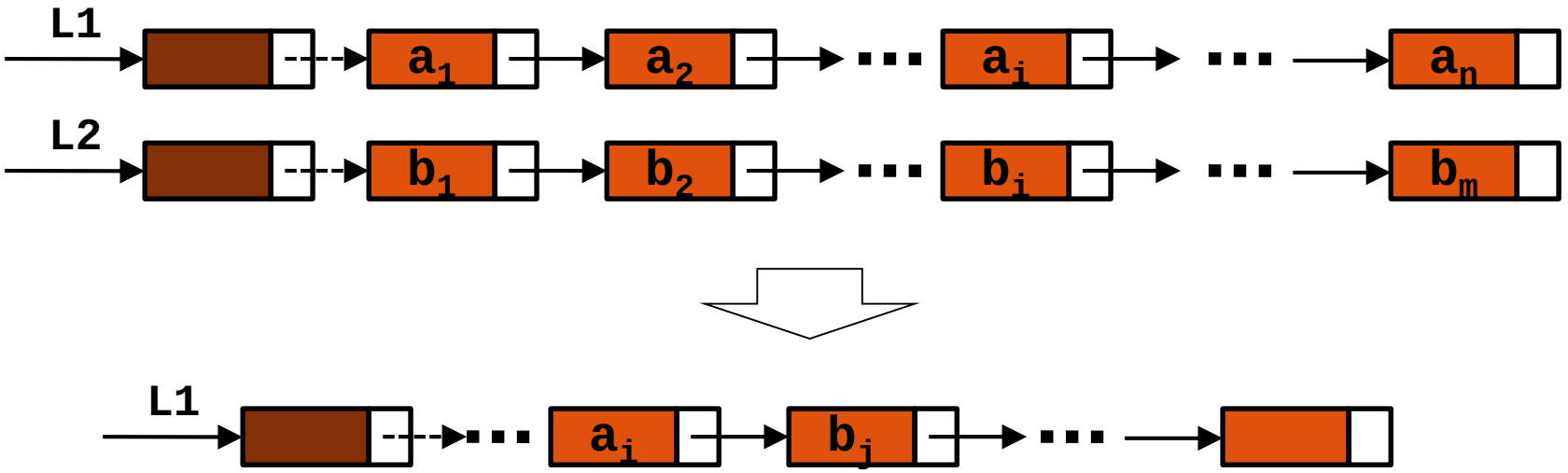


```
1.  struct node * mergenodes(struct node *L1, struct node *L2)
2.  {
3.      struct node *p1=L1, *q1=L1, *p2=L2, *q2=L2;
4.      while (p2 != NULL)
5.      {
6.              q2 = p2->next; //用q2记下p2在L2中的下一个结点
7.              while (q1!=NULL && q1->data < p2->data)
8.              {
9.                      p1 = q1; q1 = q1->next;
10.             }
11.              //如果已经到达L1表尾，可直接把L2其余部分链接到L1尾部，不需循环
12.             if (q1 == NULL)
13.                     break;
14.             if (q1 == L1) //若插入位置在L1的表头，需修改L1
15.             {
16.                     p2->next = q1;  p1 = q1 = L1 = p2;
17.             }
18.             else //若插入位置在L1的其他位置
19.             {
20.                     p2->next = q1; p1->next = p2;
21.                     p1 = p2;
22.             }
23.             p2 = q2; //p2指向L2中的下一个结点
24.     }// 未完待续
```

```
25.     //紧接上页
26.     if (q1 == NULL) //如果已经到达L1表尾，可直接把L2其余部分链接到L1尾部
27.     {
28.             if (p1 == NULL)
29.             {
30.                     L1 = p2;
31.             }
32.             else
33.             {
34.                     p1->next = p2;
35.             }
36.     }
37.
38.     return L1;
39. }
```

带头结点的有序链表的合并

- 如图所示，链表L1和L2的结点数据顺序递增，写一个函数把L2合并到L1中，且保持结点数据的顺序递增。



```
1.  struct node * mergenodes(struct node *L1, struct node *L2)
2.  {
3.      struct node *p1=L1, *q1=L1->next, *p2=L2->next, *q2=L2->next;
4.      while (p2 != NULL)
5.      {
6.              q2 = p2->next; //用q2记下p2在L2中的下一个结点
7.              while (q1!=NULL && q1->data < p2->data)
8.              {
9.                      p1 = q1; q1 = q1->next;
10.             }
11.              //如果已经到达L1表尾，可直接把L2其余部分链接到L1尾部，不需循环
12.             if (q1 == NULL)
13.                     break;
14.             p2->next = q1;
15.             p1->next = p2;
16.             p1 = p2;
17.             p2 = q2;          //p2指向L2中的下一个结点
18.     }
19.     if (q1 == NULL) //如果已经到达L1表尾，可直接把L2其余部分链接到L1尾部
20.     {
21.             p1->next = p2;
22.     }
23.     return L1;
24. }
```

接下来的安排

- 笔试前答疑（如果需要）
  - 地点：海韵校区科研（2）楼301小会议室
  - 时间： 2017年1月8日，晚19：00~21：00