第三章：中科大软院复试资料

本部分共包括三个大部分，分别是：机试、英语面试、专业课面试。

这其中机试分为三个小部分，分别是：C 语言机试、逻辑测试、性格测试。

英语面试就一个部分：英语口语交流。

专业面试就一个部分：专业素养交流。但是该部分涉猎广泛，大致可以分为十个小部分，分别是：数据结构、微机原理、计算机组成原理、操作系统、计算机网络、软件工程、数据库、信息安全、编译原理、通信原理、其他特殊面试问题集锦，编写该部分内容旨在帮助你了解科大软院专业课面试内容，但是复试题型变幻莫测，都是老师随机想出来的，因此本部分只能说具有极小的参考性和指导性。

本书大约收录了 380 个专业课面试问题，该部分只能提供面试思路亦或者面试方向，切忌把本资料当做面试圣经，其他标榜专业课面试圣经的资料亦不可采信，切记！切记！

3.1 机试

3.1.1 C 语言机试

C 语言机试解读

C 语言机试题一般来说是 20 或者 30 道题目，时间上差不多就是 30 分钟，卷子简单 30 分钟不到就写完了，卷子难，你就会做的很匆促，15 年总体反应比较简单，差不多国二水平，16 年总体反应比较难，跟 15 年的不是一个数量级，17 年总体反应比较简单，比 16 年简单，甚至比 15 年也简单，但是 18 年是什么情况，目前不得而知，因此下方准备了 12 组测试，每组难度不同，题目数量也不同，要么 20 题一组，要么 30 题一组。希望尽量能给各位模拟出真实的感觉。

C 语言机试测试卷（一）

1、有以下定义语句double a,b;int w; long c;若各变量已正确赋值，则下列选项中正确的表达式是_____。

A：a=a+b=b++　　B：w%((int)a+b)

C：(c+w)%(int)a　　D：w=a==b;

2、以下4组用户定义标识符中，全部合法的一组是_____。

①	②	③	④
_main	If	txt	int

enclude	-max	REAL	k_2
sin	turbo	3COM	_001

A：①　　B：②
C：③　　D：④

3、以下不合法的字符常量是_____。
A：'\018'　　B：'\"'
C：'\\'　　D：'\xcc'

4、表达式3.6-5/2+1.2+5%2的值是_____。
A：4.3　　B：4.8
C：3.3　　D：3.8

5、有以下程序
```
#include  <stdio.h>
main() {
    char  c1='1',c2='2';
    c1=getchar();
    c2=getchar();
    putchar(c1);
    putchar(c2);
}
```
当运行时输入：a<回车>后，以下叙述正确的是_____。
A：变量c1被赋予字符a，c2被赋予回车符
B：程序将等待用户输入第2个字符
C：变量c1被赋予字符a，c2中仍是原有字符2
D：变量c1被赋予字符a，c2中将无确定值

6、设变量均已正确定义，若要通过scanf("%d%c%d%c",&a1,&c1,&a2,&c2)；语句为变量a1和a2赋数值10和20，为变量c1和c2赋字符X和Y。以下所示的输入形式中正确的是（注：□代表空格字符）_____。
A：10□X□20□Y<回车>
B：10□X20□Y<回车>
C：10□X<回车>20□Y<回车>
D：10X<回车>20Y<回车>

7、以下合法的字符型常量是_____。
A：'\x13'　　B：'\081'
C：'65'　　D："\n"

8、以下选项中可作为C语言合法整数的是_____。
A：10110B　　B：0386
C：0Xffa　　D：x2a2

9、以下关于逻辑运算符两侧运算对象的叙述中正确的是_____。
A：只能是整数0或1　　B：只能是整数0或非0整数
C：可以是结构体类型的数据　　D：可以是任意合法的表达式

10、设有定义：int a;　float b;　　执行scanf("%2d%f",&a,&b)；语句时，若从键盘输入876　543.0<回车>，a和b的值分别是_____。
A：876和543.000000　　B：87和6.000000
C：87和543.000000　　D：76和543.000000

11、若x和y都是int型变量，x=100,y=200，且有下面的程序片段
printf("%d",(x,y));
上面程序片段的输出结果是_____。
A：200　　B：100
C：100　200　　D：输入格式符不够，输出不确定的值

12、在C语言中，合法的长整型常数是_____。
A：0L　　B：4962710
C：324562&　　D：216D

13、有以下程序
```
main()
{  int  x=102,y=012;
printf("%2d,%2d\n",x,y);
}
```
执行后输出结果是_____。
A：10,01　　B：02,12
C：102,10　　D：02,10

14、已有定义：int x=3,y=4,z=5;，则表达式!(x+y)+z-1&&y+z/2的值是_____。
A：6　　B：0
C：2　　D：1

15、在C语言中，不正确的int类型的常数是_____。
A：32768　　B：0
C：037　　D：0xAF

16、下列叙述中正确的是_____。
A：C语言中既有逻辑类型也有集合类型
B：C语言中没有逻辑类型但有集合类型
C：C语言中有逻辑类型但没有集合类型
D：C语言中既没有逻辑类型也没有集合类型

17、下列关于单目运算符++、--的叙述中正确的是_____。
A：它们的运算对象可以是任何变量和常量

B：它们的运算对象可以是char型变量和int型变量，但不能是float型变量
C：它们的运算对象可以是int型变量，但不能是double型变量和float型变量
D：它们的运算对象可以是char型变量、int型变量和float型变量

18、若有以下程序段：
```
int  m=0xabc,n=0xabc;
m-=n;
printf("%X\n",m);
```
执行后输出结果是_____。
A：0X0　　B：0x0
C：0　　D：0XABC

19、有以下程序
```
main()
{   char  a,b,c,*d;
a='\'; b='\xbc';
c='\0xab'; d="\0127";
printf("%c%c%c%c\n",a,b,c,*d);
}
```
编译时出现错误，以下叙述中正确的是_____。
A：程序中只有a='\';语句不正确
B：b='\xbc';语句不正确
C：d="\0127";语句不正确
D：a='\';和 c='\0xab';语句都不正确

20、以下程序运行时，若从键盘输入：10 20 30<回车>，输出的结果是_____。
```
#include <stdio.h>
main(){
   int i=0,j=0,k=0;
   scanf("%d%*d%d",&i,&j,&k);
   printf("%d%d%d\n",i,j,k);
}
```

C 语言机试测试卷（一）解析

1、答案：C
解析：本题考查的是表达式的赋值。
赋值运算符的左边只能是变量，不能是表达式，而选项A正是犯了此错误；选项B中类型不匹配，因为"%"两边表达式值的类型必需为整型；选项D的后面不能加"；"，因为题目要求的是正确的表达式，而表达式后面不能有"；"，只有语句后面才带"；"。故本题答案为C。

2、答案：A
解析：C语言规定标识符只能由字母、数字和下划线三种符号组成，而且第一个

字符必须是字母或下划线。
②中的-max不符合要求，'-'既不是字母、数字，也不是下划线；
③中的3COM第一个字符为数字，不合法；
④中的int和C语言的关键字重名，不合法。
故本题答案为A。

3、答案：A
解析：本题考查的是字符常量的定义。
C语言的字符常量是用单引号（即撇号）括起来的一个字符。如'a','A','?','$'等都是字符常量。注意：'a'和'A'是不同的字符常量。除了以上形式的字符常量外，C还允许用一个"\"开头的字符序列。如'\ddd'表示1到3位八进制数所代表的字符，而八进制是由0到7这几个数字组成的，所以选项A是不合法的字符常量。
故本题答案为A。
解析补充：选项 B)是表示一个双引号的转义字符表示方法；选项 C)表示的是一个反斜杆；选项 D)表示一个 ASCⅡ值为十六进制值 cc 的字符。

4、答案：D
解析：本题考查的是表达式的运算。
算术运算符的结合方向为"自左至右"，先按运算符的优先级别高低次序执行，同时在C中两个整数相除或求余数其结果依旧为整数。计算后结果为3.8。故本题答案为D。

5、答案：A
解析：本题考查的是输入输出函数。
函数getchar( )的作用是从终端（或系统隐含指定的输入设备）输入一个字符，且只能接受一个字符（回车符也算是一个字符）。故本题中变量c1被赋予字符a，c2被赋予回车符。故本题答案为A。

6、答案：D
解析：本题考查的是通过输入/输出语句的应用。
当从键盘输入字符，并且在格式说明中未指定宽度时，输入的字符（包括空格符、回车符、Tab符）将按顺序赋予各输入项。也就是说，输入的字符之间没有间格符。以选项A的形式输入数据，就是将10赋予a1，将空格赋予c1，接着a2将遇到的字母X，因为类型不匹配，虽然程序照常运行而且并不报错，但scanf函数将结束执行，使得a2和c2不能从终端接收数据。选项B、选项C存在相同的问题，选项D的输入格式正确。故本题答案选D。

7、答案：A
解析：本题考查的是字符常量。
字符常量只能包含一个字符，因此选项C错误。字符常量只能用单引号括起来，不能用双引号括起来，因此选项D错误。转义字符常量是以一个反斜线开头后跟一个特定的字符，用来代表某个特定的ASCII字符，反斜线后直接跟数字表示八进制（八进制不用0开头），所以选项B错误。反斜线后用小写字母x开头来表示

十六进制数。故本题答案选A。

8、答案：C
解析：本题的考查点是C语言中的整型常数。
整型常数可以用十进制、八进制或十六进制表示，并分为有符号数、无符号数和长整数。由此可知选项A错；
选项B，用0386表示八进制错误，因为八进制数为0-7八个数字，含有8是不对的；
选项D，数字前加"0x"表示十六进制数，所以D也不对。故本题答案为C。

9、答案：D
解析：本题考查的是C语言运算符的基本应用。
逻辑运算符两侧的运算对象不但可以是 0 和 1，也可以是任意基本类型或指针类型的数据，还可以是任意合法表达式。故本题答案为 D。
解析补充：C 语言提供了三大类数据类型，即基本类型、枚举类型、空类型、派生类型
1、基本类型
(1)整形类型：int,short int,long int,long long int(C99),char,bool
(2) 浮点类型：float,double; 双精度浮点型(float_complex,double_complex,long long_comple)
2、枚举类型 enum
3、空类型 void
4、派生类型
(1)、指针类型 *
(2)、数组类型 []
(3)、结构体类型 struct
(4)、共用体类型 union
(5)、函数类型

10、答案：B
解析：本题考查的是数据的输入输出。
对于unsigned型数据可以指定数据的输入域宽w，系统将自动按w值截取所需长度的数据，此题中指定输入域宽为2，所以把前两个数送给a，即a=87；后面的数值应送给变量b，由于6后面是空格分隔符，所以系统认为该数据到此结束，即将6赋给了b。故本题答案为B。

11、答案：A
解析：本题的考查点是逗号表达式。
在(x,y)中的","是一个特殊的运算符，叫做逗号运算符，它的一般形式为：表达式1,表达式2，求解过程为：先求解表达式1，再求解表达式2，整个表达式的值是表达式2的值，(x,y)的值为200，所以输出结果为200。故本题答案为A。

12、答案：A
解析：本题考查C语言数据类型。C语言中长整型数为在数值后加上一个L或l字符。本题答案为A。

13、答案：C
解析：y=012表示将八进制数12赋给变量y。d格式符，用来输出十进制整数。%md，m为指定的输出字段的宽度。如果数据的位数小于m，则左端补以空格，若大于m，则按实际位数输出。本题是将八进制数12也输出为十进制数，八进制数12转换为十进制数为10，所以输出结果为102,10。
故本题答案为C。

14、答案：D
解析：本题的考查点是运算符的优先级。
在本题中，运算符的优先级分别为：！大于/大于+,-大于&&，即：!(3+4)+5-1&&4+5/2=0+5-1&&4+2=4&&6=1。故本题答案为D。

15、答案：A
解析：整型（int）常量即整常数。C整常数可用以下三种形式表示：
（1）十进制整数。如123，-456，0。
（2）八进制整数。以0开头的数是八进制数。如0123表示八进制数123，即$(123)_8$，其值为：1×8^2+2×8^1+3×8^0，等于十制数83。-011表示八进制数-11，即十进制数-9。
（3）十六进制整数。以0x开头的数是16进制数。如0x123，代表16进制数123，即$(123)_{16}$=1×16^2+2×16^1+3×16^0=256+32+3=291。-0x12等于十进制数-18。
一个int型的常数的值范围为-2^15~（2^15-1），即-32768~32767。故本题选项A不是正确的int类型的常数。本题答案为A。

16、答案：D
解析：本题的考查点是C语言中的基本数据类型。
数据类型是所允许的数据及其操作的集合，是高级语言的重要特征。程序设计中可以利用数据类型发现程序中的某些错误。
C语言提供了三大类数据类型，即基本类型、复合类型和地址类型
1,基本类型只代表单个数据；
2,复合类型由基本类型组合而成，可代表一批数据；
3,地址类型可直接表示内存中的地址。
C语言支持的五种基本数据类型：
字符型：表示单个字符；
整型：表示整数，包括基本整型、短整型、长整型和无符号整型；
浮点型：表示实数，精度为6-7位有效数字；
双精度型：表示实数，精度为15-16位有效数字；
无值类型：表示无返回值的函数或无定向指针。
故本题答案为D。

17、答案：D

解析：本题的考查点是自增、自减运算符。
自增、自减运算符的作用是使变量的值增1或减1，只能用于变量，包括char型变量、int型变量和float型变量，而不能用于常量或表达式。故本题答案为D。

18、答案：C
解析：本题的考查点是不同类型数据的输入输出。
printf()函数的一般格式如下所示：
printf("格式控制字符串",输出项清单)
格式控制字符串中包括：
1.格式转换说明符：以"%"开头后跟一个格式字符，它用于输出数据格式的转换与控制；
2.转义字符：以"\"开头后跟一个字符，用于输出某些特殊意义的字符和不可显示字符；
3.其他字符：用于照原样显示的字符。printf()格式转换说明符："%X"按实际位数输出十六进制整数。本题因为m，n都定义为十六进制整数，且m-=n;的意思是将m与n的差值重新赋给m，所以m为0。故本题答案为C。

19、答案：D
解析：本题的考查点是字符常量。
C 中的字符常量是用单引号（即撇号）括起来的一个字符。如'a'、'x'、'D'、'?'、'$'等都是字符常量。除了以上形式的字符常量外，C 还允许用一种特殊形式的字符常量，就是以一个"\"开头的字符序列。'\ddd'表示 1 到 3 位 8 进制数所代表的字符，不足 3 位就在前面加 0；'\xhh'表示 1 到 2 位 16 进制数所代表的字符。而 a='\';和 c='\0xab';是不正确的。故本题答案为 D。

20、答案：10300
解析：本题考查的是格式控制符。
在scanf格式控制符中，如果在%后有一个"*"附加说明符，表示跳过它指定的列数。本题中""%d%*d%d""表示将10赋给i，%*d表示读入整数但不赋给任何变量，然后再读入整数30赋给变量j，那么变量k并没有重新赋值，仍为初始值0。所以输出的结果为10300。

C 语言机试测试卷（二）

1、若有语句
```
int i=-19,j=i%4;
printf("%d\n",j);
```
则输出的结果是_____。

2、设有以下语句
```
char a=3,b=6,c;
c=a^b<<2;
```
则c的二进制值是_____。

A：00011011　　B：00010100
C：00011100　　D：00011000

3、有以下程序

```
main()
{   int x=3, y=2, z=1;
printf("%d\n",x/y&~z);
}
```

程序运行后的输出结果是_____。

A：3　　B：2　　C：1　　D：0

4、以下程序的功能是进行位运算

```
main()
{   unsigned char  a,b;
a=7^3;  b=~4&3;
printf("%d  %d\n",a,b);
}
```

程序运行后的输出结果是_____。

A：4　3　　B：7　3
C：7　0　　D：4　0

5、设有如下程序

```
#include<stdio.h>
main()
{
int **k, *j,i=100;
j=&i;    k=&j;
printf("%d\n",**k);
}
```

上述程序的输出结果是_____。

A：运行错误　　B：100
C：i的地址　　D：j的地址

6、若定义了以下函数：

```
void f(……)
{……
*p=(double*)malloc(10*sizeof(double));
……
}
```

p是该函数的形参，要求通过p把动态分配存储单元的地址传回主调函数，则形参p的正确定义应当是_____。

A：double *p　　B：float **p
C：double **p　　D：float *p

7、有以下程序

```
#include <stdlib.h>
main()
{   char *p,*q;
p=(char *)malloc(sizeof(char)*20); q=p;
scanf("%s%s",p,q); printf("%s%s\n",p,q);
}
```

若从键盘输入：abc def↙，则输出结果是_____。

A：defdef　　B：abcdef

C：abcd　　D：dd

8、有以下程序

```
#include <stdio.h>
main()
{   int a[]={1,2,3,4,5,6,7,8,9,10,11,12,},*p=a+5,*q=NULL;
*q=*(p+5);
printf("%d %d\n",*p,*q);
}
```

程序运行后的输出结果是_____。

A：运行后报错　　B：6 6

C：6 11　　D：5 10

9、有以下程序

```
main()
{   char s[]="Yes\n/No",*ps=s;
puts(ps+4);
*(ps+4)=0;
puts(s);
}
```

程序运行后的输出结果是_____。

A：n/No
Yes
/No

B：/No
Yes

C：/NO
Yes
/No

D：
/No
Yes

10、在以下语句中存在语法错误的是_____。

A：char ss[6][20]; ss[1]="wrong?";

B：char ss[][20]={"wrong?"};

C：char *ss[6]; ss[1]="wrong?";

D：char *ss[]={"wrong?"};

11、若有下面的说明和定义，则sizeof(struct aa)的值是_____。

```
struct aa
{   int r1;double r2;float r3;
    union uu{
          char u1[5];
         long u2[2];
     }ua;
}maya;
```

A：30　　B：29　　C：24　　D：32

12、以下选项中不能正确把c1定义成结构体变量的是_____。

```
1,typedef struct
{int red;
int green;
int blue;
} COLOR;
COLOR c1;
```

```
2,struct color c1
{ int red;
 int green;
 int blue;
 };
```

```
3,struct color
{ int red;
int green;
int blue;
}c1;
```

```
4,struct
{int red;
int green;
int blue;
}c1;
```

A：1　　B：2　　C：3　　D：4

13、若有以下说明和定义

```
union dt
{int a;char b;double c;} data;
```

以下叙述中错误的是_____。

A：data的每个成员起始地址都相同

B：变量data所占的内存字节数与成员c所占字节数相等

C：程序段：data.a=5;printf("%f\n",data.c);输出结果为5.000000

D：data可以作为函数的实参

14、有以下程序

```
typedef struct{int b,p;}A;
void f(A c)                    /* 注意：c是结构变量名 */
{   int j;
c.b+=1; c.p+=2;
}
main()
{   int i;
A a={1,2};
```

```
f(a);
printf("%d,%d\n",a.b,a.p);
}
```
程序运行后的输出结果是_____。

15、若有定义：float x=1.5;int a=1,b=3,c=2;则正确的switch语句是_____。

A：
```
switch(x)
{
case 1.0: printf("*\n");
case 2.0: printf("**\n");
}
```

B：
```
switch((int)x);
{
case 1: printf("*\n");
case 2: printf("**\n");
}
```

C：
```
switch(a+b)
{
case 1: printf("*\n");
case 2+1: printf("**\n");
}
```

D：
```
switch(a+b)
{
 case 1: printf("*\n");
 case C: printf("**\n");
 }
```

16、若a,b,c1,c2,x,y均是整型变量，正确的switch语句是_____。

①
```
switch(a+b);
{   case 1:y=a+b;break;
case 0:y=a-b;break;
}
```

②
```
switch(a*a+b*b)
{   case 3:
case 1:y=a+b;break;
case 3:y=b-a;break;
}
```

③
```
switch  a
{   case c1:y=a-b;break;
case c2:x=a*b;break;
default:x=a+b;
}
```

④
```
switch (a-b)
{   default:y=a*b;break;
 case  3:case 4:x=a+b;break;
 case 10:case 11:y=a-b;break;
 }
```

A：①　　B：②

C：③　　D：④

17、下面程序的输出是_____。
```
main( )
{   int x=3 ,y=6,a=0;
while(x++!=(y-=1) )
{   a+=1;
if (y<x) break;
}
printf("x=%d,y=%d,a=%d\n",x,y,a);
}
```

A：x=4,y=4,a=1　　B：x=5,y=5,a=1

C：x=5,y=4,a=3　　D：x=5,y=4,a=1

18、以下程序的输出结果是_____。

```
main()
{  int    x＝10,y＝10,i;
for(i＝0;x>8 ;y＝++i)
printf("%d   %d  ",x--,y);
}
```

A：10 1 9 2　　B：9 8 7 6

C：10 9 9 0　　D：10 10 9 1

19、有如下程序

```
main()
{     int    a = 2,b =- 1,c = 2;
if(a<b)
if(b<0)  c=0;
else  c++;
printf("%d\n",c);
}
```

该程序的输出结果是_____。

A：0　　B：1　　C：2　　D：3

20、s1和s2已正确定义并分别指向两个字符串。若要求：当s1所指串大于s2所指串时，执行语句S;，则以下选项中正确的是_____。

A：if(s1>s2) S;

B：if(strcmp(s1,s2)) S;

C：if(strcmp(s2,s1)>0) S;

D：if(strcmp(s1,s2)>0) S;

C 语言机试测试卷（二）解析

1、答案：-3

解析：本题的考查点是求模运算。

算术运算符中，%只能对整型运算量施加运算。在计算两个整数的余数时，余数的符号与被除数相同。例如，3%(-2)=1，-3%2=-1，3%2=1，-3%(-2)=-1。故本题答案为：-3。

2、答案：A

解析：本题的考查点是位运算。

"<<"是C语言中规定的左移运算符，例如，a=a<<2，这个语句即是将a的二进制数左移两位，左移一位相当于该数乘以2，左移两位相当于该数乘以2的2次方；^是异或运算符，所以，c的二进制值应为00011011。故本题答案为A。

3、答案：D
解析：本题的考查点是位运算符的知识点。
&是"按位与"运算符，参加运算的两个运算量的相应位都为1，则该位的结果值为1，否则为0。~是"取反"运算符，用来对一个二进制数按位取反，即将0变1，1变0。本题x/y&~z，x/y的值为1，~z的值为0，所以进行&运算后，整个表达式的值为0。故本题答案为D。
补充解析：&是一个位运算符，就是将两个二进制的数逐位相与，就是都是1才是1，只要有一个为0则为0，结果是相与之后的结果。
&&是一个逻辑运算符，就是判断两个表达式的真假性，只有两个表达式同时为真才为真，有一个为假则为假。原题如果改成&&，答案为1.

4、答案：A
解析：本题考查的是位运算。C语言提供六种位运算符，按优先级由高到低的顺序分别为：取反（～）→左移（<<）和右移（>>）→按位与（&）→按位异或（∧）→按位或（|）。"∧"是按位异或运算，当对应位上的二进制数值相同时，异或的结果为0，对应位上的二进制数值不同时，异或的结果为1，本题"7∧3"相当于"0111∧0011"，所以a的值为4；"&"是按位与运算符，只有当所有对应位上的数值都为1时，结果才为1；"~"是按位取反。题中"~4&3"相当于1011&0011=0011，所以b的值为3。 故本题答案为A。

5、答案：B
解析：本题的考查点是指针变量的引用。
j=&i，j的值就是i的地址，*j=100，再将j的地址赋给k，这时*k=j，那么，**k=*j，而*j=100，所以**k=100，最后的打印结果应当为100。故本题答案为B。

6、答案：C
解析：通过程序中给出的语句，我们可以判断，*p是一个指向double型指针的指针。因此，要通过p把动态分配存储单元的地址传回主调函数，应该使用double **p。本题答案为C。

7、答案：A
解析：本题的考查点是malloc()函数的应用。
malloc()是在内存的动态存储区中分配一个长度为size的连续空间。此函数的值（即"返回值"）是一个指针，它的值是该分配域的起始地址。如果此函数未能成功地执行，则返回值为0。本题只开辟了一片连续的存储单元，只能存储一个字符串的值，字符串遇到空字符时即结束，当输入第二个字符串时将第一个字符串覆盖，最后只打印出第二个串。故本题答案为A。

8、答案：A
解析：此题考查的是指针变量赋值。
本题将指针q赋值为空，即指向了空地址，而对空地址所对应的内容赋值*q=*(p+5)是会出错的，所以输出结果会提示Null pointer assignment。所以此题答案为A。

9、答案：B
解析：此题考的是puts()函数。
其作用是将一个字符串（以'\0'结束的字符序列）输出到终端。用puts函数输出的字符串中可以包含转义字符，遇到转义字符，自动跳过去。例如本题中ps+4指的是'/'的地址，并不是'\n'的地址。所以puts(ps+4)输出的是/NO。
C规定：在每一个字符串的结尾加一个"字符串结束标志"，以便系统据此判断字符串是否结束。C规定以字符'\0'作为字符串结束标志。'\0'是一个ASCII码为0的字符，从ASCII代码表中可以看到ASCII码为0的字符是"空操作字符"，即它不引起任何控制动作，也不是一个可显示的字符。所以本题执行*(ps+4)=0之后，再执行puts(s)，就相当于输出字符串Yes后遇到结束标志。故本题答案为B。

10、答案：A
解析：本题考查的是字符数组和字符指针变量的区别。
字符数组由若干个元素组成，每个元素中放一下字符，而字符指针变量中放的是地址（字符串的首地址），决不是将字符串放到指针变量中，选项D是指在定义字符指针变量时让字符指针指向字符串"wrong?"的首地址；选项C是先定义字符指针数组，再让指针数组指向字符串"wrong?"的首地址；选项B是在定义数组时将字符赋给数组，而选项A的这种先定义，再用赋值语句赋值的形式是错误的，因为数组可以在变量定义时整体赋初值，而不能在赋值语句中整体赋值。故本题答案为A。

11、答案：D
解析：答案：32
解析：（根据编译器结果为32）

C/C++中不同数据类型所占用的内存大小

	32 位	64 位
char	1	1
int	4	大多数 4，少数 8
short	2	2
long	4	8
float	4	4
double	8	8
指针	4	8

（单位都为字节）

结构体(struct)：比较复杂，对齐问题。

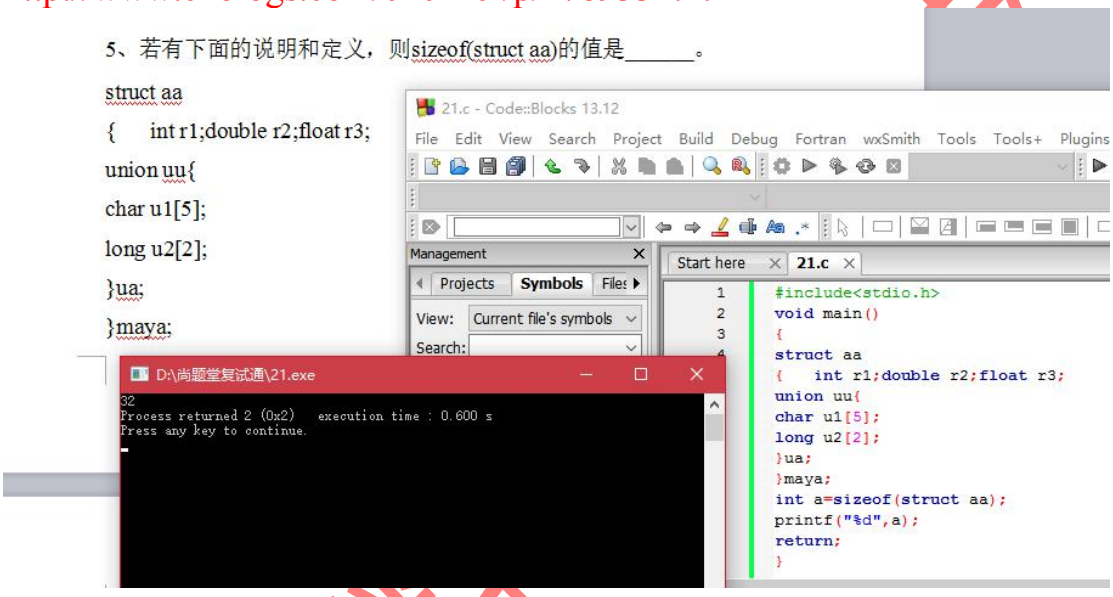
联合(union)：所有成员中最长的。

枚举(enum)：根据数据类型。

long 型的变量需要 4 个字节，long u2[2]即需要 8 个字节，所以共用体 uu 需要 8 个字节的存储空间。结构体的大小不是所有成员大小简

单的相加，需要考虑到系统在存储结构体变量时的地址对齐问题。对于嵌套的结构体，需要将其展开。对结构体求 sizeof 时，原则为：（1）展开后的结构体的第一个成员的偏移量应当是被展开的结构体中最大的成员的整数倍。（2）结构体大小必须是所有成员大小的整数倍，这里所有成员计算的是展开后的成员，而不是将嵌套的结构体当做一个整体。则 sizeof(struct aa)的值为 8*4=32.

（以下为编译结果和参考网址）
http://www.cnblogs.com/0201zcr/p/4789332.html



12、答案：B
解析：本题的考查点是结构体变量的定义。
将一个变量定义为标准类型与定义为结构体类型不同之处在于：后者不仅要求指定变量为结构体类型，而且要求指定为某一特定的结构体类型(例如，struct color)，不能只指定结构体名。其一般形式为"
struct 结构体名
{
成员表列
}变量名表列;
其中可以不出现结构体名，选项D就是缺省结构体名的情况。而变量名列表必须放在成员列表后面，所以B选项不能正确将c1定义为结构体变量。故本题答案为B。

13、答案：C
解析：本题考查的是共用体。
union是表示共用体的关键字，说明其内的成员a,b,c共占用同一个内存空间，所以data的每个成员起始地址都相同；共用体变量所占的内存长度等于最长的成员的长度，故变量data所占的内存字节数与成员c所占字节数相等；但由于各成员变量的类型不一致，所以它们的存储方式也不相同，整型占用2个字节的存储空间，double型占用8个字节的存储空间；另外data可以作为函数的实参，此时data

作为一个地址进行地址传送；执行"data.a=5;printf("%f\n",data.c);"系统不能自动将整型变量转换为浮点型值,printf函数只是将内存中的数据以不同的类型输出,而不能将内存中的整型数据自动转换为等值的浮点数,故C是错误的。故本题答案为C。

14、答案：1,2
解析：本题考查的是变量作为函数的参数和函数的调用。
C语言规定，在函数调用时变量作为参数传递是进行值传递，在函数体中对形参变量的修改不会影响对应的实参变量。本题定义了一个函数f()，用变量作为形参，在main()函数中调用了一次f()函数，形参进行计算：c.b+=1=1+1=2，c.p+=2=2+2=4，由于变量作为参数进行传递不会影响到实参的的值，所以a.b，a.p依旧是原来的值，即：1,2。故本题答案1,2。

15、答案：C
解析：本题考查的是switch语句的应用。
在switch语句中，case后的常量表达式只能是整型或等价于整型的常量，而选项A中case后面的表达式为浮点型，选项D中case后面的表达式存在字符变量"c"，故选项A与选项D错误；在选项B中，switch后面的表达式加了分号显然格式有误。故本题答案为C。

16、答案：D
解析：选项A的错误在于switch语句后面不该加分号；
选项B的错误在于switch语句要求每一个case的常量表达式的值必须互不相同；
选项C的错误在于变量a应该用括号括起来，case后面应该是常量表达式，而不是变量。本题答案为D。

17、答案：D
解析：本题考查的是运算符。
注意的是X++中X值应当是先引用，后自加，搞清楚这一点以后，只要逐步将值带入程序中，即可得到正确的选择。故本题答案为D。

18、答案：D
解析：本题考查的是for循环，第一次循环x=10，y=10，输出结果为10　10，第二次循环x=9,y=1，输出结果为9　1。故本题答案为D。
补充解析：++i为先对i加1，再进行表达式计算。第一次循环时x=10，y=10，打印结束后x=9，第二次循环结束后y=1。第二次循环时x=9，y=1。

19、答案：C
解析：由于条件(a < b)并不满足，所以if(a < b)后面的语句并没有被执行，c仍然等于2。本题答案为C。

20、答案：D
解析：本题考的是字符串比较函数strcmp(char *s1,char *s2)。
字符串比较规则是对两个字符串自左至右逐个字符相比（按 ASCII 码值大小比

较），直到出现不同的字符或遇到"\0"为止。如果全部相同，则认为相等；若出现不相同的字符，则以第一个不相同的字符的比较结果为准。比较的结果由函数值带回。当 s1<s2，返回值<0；当 s1=s2 时，返回值=0；当 s1>s2 时，返回值>0；所以在此题中要使得 s1 所指串大于 s2 所指串，那么就必定使得 strcmp(s1,s2)>0。故本题答案为 D。

C 语言机试测试卷（三）

1、已有定义：char a[]="xyz",b[]={'x','y','z'};，以下叙述中正确的是_____。

A：数组a和b的长度相同

B：a数组长度小于b数组长度

C：a数组长度大于b数组长度

D：上述说法都不对

2、以下叙述中错误的是_____。

A：对于double类型数组，不可以直接用数组名对数组进行整体输入或输出

B：数组名代表的是数组所占存储区的首地址，其值不可改变

C：当程序执行中，数组元素的下标超出所定义的下标范围时，系统将给出"下标越界"的出错信息

D：可以通过赋初值的方式确定数组元素的个数

3、若有定义：char *x="abcdefghi";，以下选项中正确运用了strcpy函数的是_____。

A：char y[10]; strcpy(y,x[4]);

B：char y[10]; strcpy(++y,&x[1]);

C：char y[10],*s; strcpy(s=y+5,x);

D：char y[10],*s; strcpy(s=y+1, x+1);

4、在C语言中，函数中变量的隐含存储类别是_____。

A：auto　　B：static

C：extern　　D：无存储类别

5、下列叙述中正确的是_____。

A：C语言编译时不检查语法

B：C语言的子程序有过程和函数两种

C：C语言的函数可以嵌套定义

D：C语言中，函数根据其能否被其它源文件调用，分为内部函数和外部函数

6、以下存储类型中，只有在使用时才为该类型的变量分配内存的是_____。

A：auto和static　　B：auto和register

C：register和static　　D：extern和register

7、在调用函数时，如果实参是简单变量，它与对应形参之间的数据传递方式是_____。

A：地址传递
B：单向值传递
C：由实参传给形，再由形参传回实参
D：传递方式由用户指定

8、以下函数值的类型是_____。

```
fun(float x)
{   float y;
y=3*x-4;
return y;
}
```

A：int　　B：不确定
C：void　　D：float

9、以下叙述中正确的是_____。
A：局部变量说明为static存储类，其生存期将得到延长
B：全局变量说明为static存储类，其作用域将被扩大
C：任何存储类的变量在未赋初值时，其值都是不确定的
D：形参可以使用的存储类说明符与局部变量完全相同

10、程序中对fun函数有如下说明

```
void *fun();
```

此说明的含义是_____。
A：fun函数无返回值
B：fun函数的返回值可以是任意的数据类型
C：fun函数的返回值是无值型的指针类型
D：指针fun指向一个函数，该函数无返回值

11、有以下程序

```
fun(int  x, int  y){return(x+y);}
main()
{   int  a=1,b=2,c=3,sum;
sum=fun((a++,b++,a+b),c++);
printf("%d\n",sum);
}
```

执行后的输出结果是_____。
A：6　　B：7　　C：8　　D：9

12、有以下程序

```
int fun1(double a){return a*=a;}
int fun2(double x,double y)
{   double a=0,b=0;
a=fun1(x);b=fun1(y); return(int)(a+b);
}
```

```
main()
{double w;w=fun2(1.1,2.0);……}
```
程序执行后变量w中的值是_____。

A：5.21
B：5
C：5.0
D：0.0

13、有以下程序

```
int fun1(double a){return a*=a;}
int fun2(double x,double y)
{   double a=0,b=0;
a=fun1(x);b=fun1(y); return(int)(a+b);
}
main()
{double w;w=fun2(1.1,2.0);……}
```
程序执行后变量w中的值是_____。

A：5.21
B：5
C：5.0
D：0.0

14、若有以下函数首部

```
int  fun(double  x[10], int  *n)
```
则下面针对此函数的函数声明语句中正确的是_____。

A：int fun(double x, int *n);
B：int fun(double, int);
C：int fun(double *x, int n);
D：int fun(double *, int *);

15、有以下程序

```
void f(int n,int *r)
{   int r1=0;
if(n%3==0)   r1=n/3;
else   if(n%5==0)  r1=n/5;
else   f(--n,&r1);
*r=r1;
}
main()
{ int m=7,r;
f(m,&r);    printf("%d\n",r);
}
```
程序运行后的输出结果是_____。

A：2

B：1
C：3
D：0

16、有以下程序

```
#include <stdio.h>
#include <string.h>
void  fun( char  s[][10],int  n)
{   char  t;   int  i,j;
for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)          /* 比较字符串的首字符大小，并交换字符串的首字符 */
if(s[i][0]>s[j][0]) {t=s[i][0];s[i][0]=s[j][0];s[j][0]=t;}
}
main()
{   char  ss[5][10]={"bcc","bbcc","xy","aaaacc","aabcc"};
fun(ss,5);    printf("%s,%s\n",ss[0],ss[4]);
}
```

程序的运行结果是_____。

A：xy,aaaacc
B：aaaacc,xy
C：xcc,aabcc
D：acc,xabcc

17、有以下程序

```
void f(int *x,int *y)
{
int t;
t=*x;*x=*y;*y=t;
}
main()
{
int a[8]={1,2,3,4,5,6,7,8},i,*p,*q;
p=a;q=&a[7];
while(p<q)
{f(p,q);p++;q--;}
for(i=0;i<8;i++) printf("%d,",a[i]);
}
```

程序运行后的输出结果是_____。

A：8,2,3,4,5,6,7,1,
B：5,6,7,8,1,2,3,4,
C：1,2,3,4,5,6,7,8,
D：8,7,6,5,4,3,2,1,

18、已定义以下函数
```
fun(char *p2, char *p1)
{while((*p2=*p1)!='\0'){p1++;p2++;} }
```
函数的功能是_____。

A：将p1所指字符串复制到p2所指内存空间
B：将p1所指字符串的地址赋给指针p2
C：对p1和p2两个指针所指字符串进行比较
D：检查p1和p2两个指针所指字符串中是否有'\0'

19、以下程序中函数f的功能是将n个字符串，按由大到小的顺序进行排序。
```
#include <string.h>
void f(char p[][10],int n)
{   char t[20]; int i,j;
for(i=0;i<n-1;i++)
for (j=i+1;j<n;j++)
if(strcmp(p[i],p[j])<0)
{ strcpy(t,p[i]);strcpy(p[i],p[j]);strcpy(p[j],t);}
}
main()
{   char  p[][10]={"abc","aabdfg","abbd","dcdbe","cd"};int i;
f(p,5); printf("%d\n",strlen(p[0]));
}
```
程序运行后的输出结果是_____。

A：6
B：4
C：5
D：3

20、设有以下函数：
```
f(int a)
{   int  b=0;
static int c=3;
b++;c++;
return(a+b+c);
}
```
如果在下面的程序中调用该函数，则输出结果是_____。
```
main()
{  int  a=2, i;
for(i=0;i<3;i++) printf("%d\n",f(a));
}
```

A)	B)	C)	D)
7	7	7	7
8	9	10	7

9　　11　　13　　7

A：A)
B：B)
C：C)
D：D)

C 语言机试测试卷（三）解析

1、答案：C
解析：本题考查的是数组定义。
C语言规定'\0'为字符串结束标志，系统对字符串常量自动加一个'\0'为结束符。所以"char a[]="xyz""的数组长度为4，而"b[]={'x','y','z'};"的数组长度为3（此处的数组长度与strlen函数所求的长度不同，此处的长度是指数组占内存空间的大小），故a数组长度大于b数组长度。故本题答案为C。

2、答案：C
解析：本题考查的是数组的概念。
在C语言中，如果出现下标越界的情况，系统不管在编译还是执行时都不会给出"下标越界"的错误提示。故本题答案为C。

3、答案：D
解析：本题考查的是用于字符串处理的函数。
选项A中x[4]是取字符e，也就是将字符复制到y中，strcpy实现的是地址的复制所以选项A错误；选项B中++y是错误的，不允许对常量进行自加运算（y是一个确定的地址值），所以选项B 错误；选项C指针变量s指向了y向后的第5位，则存放x时会出现越界问题（补充：x有9位，从y+5的地址开始存放会越界），所以选项C错误；选项D中，指针变量s指向了y向后的第一位，此时s可存放数据的长度为9，而地址"x+1"起的字符串的地址长度也刚好为9（包括"\0"），所以开始复制不会出现地址越界问题。故本题答案为D。

4、答案：A
解析：本题考查的是函数中变量的存储类别。函数中变量的存储类别包括auto，static，extern，其中auto为隐含存储类别，static为静态存储类别，register是寄存器存储类别。故本题答案为A。

5、答案：D
解析：选项A的错误在于编译过程中是检查语法的，若发现源程序有语法错误，则系统会提示出错信息；选项B的错误在于C语言中，子程序的作用是由函数来完成的，无过程的概念；选项C的错误在于函数不可以嵌套定义，但可以嵌套调用。本题答案为D。

6、答案：B
解析：extern、register、static、auto分别是定义外部变量、寄存器变量、静

态变量、自动变量，其中，自动变量和寄存器变量属于动态存储，调用时临时分配单元；而静态变量和外部变量属于静态存储，在整个程序运行时间都存在。本题答案为B。

7、答案：B
解析：本题考查C语言函数的参数的传值方式。
C语言函数中的参数传递方式是按值传递，指将实参的值复制一份传递给形参，形参的改变而不影响实参，即为单向传递。本题答案为B。

8、答案：A
解析：本题考查C语言函数的缺省的函数类型。
C语言中，如果函数前不加任何数据类型时，缺省函数的类型为整型，函数的类型就是函数返回值的类型。本题答案为A。

9、答案：A
解析：此题考的是变量的作用域。
在一个函数内部定义的变量是局部变量，它只在本函数范围内有效，而全局变量的作用域为整个源程序文件，即它可以为本文件中所有函数所共用，static类型的变量是静态变量，它的值在函数调用结束后不会消失，而是保留原值，即占用存储单元不释放，由于全局变量的存储方式也是如此，即全局变量和说明为static类型的变量的作用域是相同的，所以B答案这句话是错误的，A答案是正确的；C答案也是错误的，因为对静态局部变量来说，编译时自动赋初值0（对数值型变量）或空字符（对字符型变量），而对自动变量来说，如果不赋初值则它的值是一个不确定的值，这是因为每次函数调用结束后存储单元已释放，下次调用时又重新另分配存储单元，而所分配的单元中的值是不确定的。故本题答案选A。

10、答案：C
解析：本题考的是函数的返回值。
由于()的优先级高于*，所以void *fun()表示声明了一个函数，这个函数的返回值是指向无值型变量的指针，所以本题答案为C。D答案的表达法相对应的函数说明应为void (*fun)()。故本题答案为C。

11、答案：C
解析：本题考查的是函数的正确调用。
在 fun((a++,b++,a+b),c++) 中，先算括号内的(a++,b++,a+b) 逗号运算，即先算出 a++的值，a=2；再算出 b++的值，b=3；最后算出 a+b 的值，a+b=2+3=5，整个逗号表达式的值为最后 a+b 的值即为 5；接下来调用 fun 函数，此时原语句变为：sum=fun(5,c++)；进行运算，c++表示在使用 c 以后再将 c 的值加 1，所以结果为：5+3=8。 故本题答案为 C。

12、答案：C
解析：本题考查的是数值类型的定义与强制转换。
本题在进行调用函数fun2()时，又对函数fun1()进行了调用，由于函数fun1()定义为整型，所以第一次调用fun1()时返回的值为1，然后再将1赋值给a，此时由于a为double型，则系统自动类型转换，将1转换成了1.000000赋值给了变量a，

以此类推，得b的值为4.000000，然后通过"return(int)(a+b);"将a+b的值强制转换为int型5返回给主函数，此时又由w的类型为double，所以返回的整型值5又被转换为double型。故输出结果为5.0。故本题答案为C。

13、答案：C
解析：本题考查的是数值类型的定义与强制转换。
本题在进行调用函数fun2()时，又对函数fun1()进行了调用，由于函数fun1()定义为整型，所以第一次调用fun1()时返回的值为1，然后再将1赋值给a，此时由于a为double型，则系统自动类型转换，将1转换成了1.000000赋值给了变量a，以此类推，得b的值为4.000000，然后通过"return(int)(a+b);"将a+b的值强制转换为int型5返回给主函数，此时又由w的类型为double，所以返回的整型值5又被转换为double型。故输出结果为5.0。
故本题答案为C。

14、答案：D
解析：本题考查的是函数的声明。
函数声明可以照写已定义的函数的首部，再加一个分号就成为了对函数的声明，在函数声明中也可以不写形参名，而只写形参的类型，但要保证与函数首部写法上的一致，即函数类型、函数名、参数个数、参数类型和参数顺序相同。字符数组可用指针来表示，所以选项D正确。
故本题答案为D。

15、答案：A
解析：本题考查的是指针作为函数的参数和函数的调用。
题目中定义了一个指针变量r作为形参的f()函数。在主函数main()中定义了两个变量m和r，同时给m赋初值7，执行"f(m,&r);"语句，调用f()函数并将m的初值7传递给形参n，将r的地址传给形参指针变量r。f()函数中通过指针r将最后的值返回。首先执行if语句中的条件：n%3==0，7%3=1条件为假，执行第一个else下的语句，同样不满足条件，再执行第二个else下的语句，n自行减一，n=6，再执行if语句，满足条件，r1=2，把r1的值2赋给指针变量r所指向的存储单元，即为主函数r的值，输出结果为2。
故本题答案选A

16、答案：D
解析：本题的考查点是循环嵌套。
函数fun的功能是对数组中的每个字符串的第一个字符按由小到大的顺序排序，最终输出数组的第一个和第五个的字符串。比较各字符串的首字符可知，字符'a'最小，字符'x'最大，所以输出的ss[0]的值为acc，ss[4]的值为xabcc。
故本题答案为D。

17、答案：D
解析：本题考查的是函数的调用。
本程序中函数f的作用是交换指针变量x和y所指向的存储单元的值。在主函数中，能过"p=a;q=&a[7];"将指针变量p指向了数组的第一个元素，指针变量q指向了数

组的最后一个元素。通过"f(p,q)"调用函数f进行对称的数组元素的交换。当while循环4次（即p=q）后，a数组中各元素的值分别为87654321，所以程序输出的结果为8,7,6,5,4,3,2,1,。
故本题答案为D。

18、答案：A
解析：本题的考查点是函数。
while((*p2=*p1)!='\0'表示将p1所指向的值赋给*p2，即将p1所指字符串复制到p2所指内存空间。在这里"*P2=*p1"是赋值操作，而不是比较是否相等，比较应当为双等号"=="。所以B，C，D都不对。
故本题答案为A。

19、答案：C
解析：本题的考查点是比较字符串的大小。
比较字符串的大小是从字符串的第一个字母开始比较，如果第一个字母相同则比较第二个字母，以此类推，直至字符串结束，结合本题可知，比较后的字符串数组为：
p[0]="dcdbe"
p[1]="cd"
p[2]="abc"
p[3]="abbd"
p[4]="aabdfg"
所以，strlen(p[0])=5。
故本题答案为C。

20、答案：A
解析：本题考查的是C语言的静态数据类型。静态数据类型的变量的值在退出了函数时依然存放在内存中，且只初始化一次。
本题答案为A。

C 语言机试测试卷（四）

1、下列程序的运行结果是_____。

```
void fun(int    *a,int    *b){
      int    *k;
      k=a;a=b;b=k;
}
main() {
      int a=3,b=6,*x=&a,*y=&b;
      fun(x,y);
      printf("%d %d",a,b);
}
```

A：6 3
B：3 6
C：编译出错

D：0 0

2、有以下程序

```
main(){
int a[][3]={{1,2,3},{4,5,0}},(*pa)[3],i;
pa=a;
for(i=0;i<3;i++)
if(i<2)    pa[1][i]=pa[1][i]-1;
else      pa[1][i]=1;
printf("%d\n",a[0][1]+a[1][1]+a[1][2]);
}
```

执行后输出结果是_____。

A：7
B：6
C：8
D：无确定值

3、有以下程序

```
main()
{char str[][10]={"China","Beijing"},*p=str;
printf("%s\n",p+10);
}
```

程序运行后的输出结果是_____。

A：China
B：Bejing
C：ng
D：ing

4、若有以下说明和语句

```
int c[4][5],(*p)[5];
p=c;
```

能够正确引用c数组元素的是_____。

A：p+1
B：*(p+3)
C：*(p+1)+3
D：*(p[0]+2)

5、若要用下面的程序片段使指针变量p指向一个存储整型变量的动态存储单元

```
int    *p;
p= _____ malloc(sizeof(int) );
```

则应填入_____。

A：int
B：int *
C：(*int)
D：(int*)

6、若有以下说明和语句，请选出哪个是对c数组元素的正确引用_____。

```
int c[4][5],(*cp)[5];
cp=c;
```

A：cp+1
B：*(cp+3)
C：*(cp+1)+3
D：*(*cp+2)

7、有如下程序

```
#define    N      2
#define    M    N+1
#define    NUM    2*M+1
main()
{      int    i;
for(i=1;i<=NUM;i++)printf("%d\n",i);
}
```

该程序中的for循环执行的次数是_____。

A：5
B：6
C：7
D：8

8、有以下程序

```
main()
{ int x[]={1,3,5,7,2,4,6,0},i,j,k;
for(i=0;i<3;i++)
for (j=2;j>=i;j--)
if(x[j+1]>x[j]){    k=x[j];x[j]=x[j+1];x[j+1]=k;}
for (i=0;i<3;i++)
for(j=4;j<7-i;j++)
if(x[j]>x[j+1]){ k=x[j];x[j]=x[j+1];x[j+1]=k;}
for (i=0;i<8;i++) printf("%d",x[i]);
printf("\n");
}
```

程序运行后的输出结果是_____。

A：75310246
B：01234567
C：76310462
D：13570246

9、以下程序的输出结果是_____。

```
#define    SQR(X)    X*X
main()
{    int a=16,k=2,m=1;
a/=SQR(k+m)/SQR(k+m);
printf("%d\n",a);
```

}

A：16

B：2

C：9

D：1

10、若定义了以下函数：

```
void f(……)
{……
*p=(double*)malloc(10*sizeof(double));
……
}
```

p是该函数的形参，要求通过p把动态分配存储单元的地址传回主调函数，则形参p的正确定义应当是_____。

A：double *p

B：float **p

C：double **p

D：float *p

11、以下定义语句中，错误的是_____。

A：int a[]={1,2};

B：char *a[3];

C：char s[10]="test";

D：int n=5,a[n];

12、以下程序的输出结果是_____。

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

A：4

B：5

C：6

D：0

13、若有说明语言：int a,b,c,*d=&c;则能正确从键盘读入三个整数分别赋给变量a，b，c的语句是_____。

A：scanf("%d%d%d",&a,&b,d);

B：scanf("%d%d%d",&a,&b,&d);
C：scanf("%d%d%d",a,b,d);
D：scanf("%d%d%d",a,b,*d);

14、以下程序的输出结果是_____。

```
main()
{int a=0,i;
for(i=1;i<5;i++)
{   switch(i)
{case0:
case3:a+=2;
case1:
case2:a+=3;
default:a+=5;
}
}
printf("%d\n",a)
}
```

A：31
B：13
C：10
D：20

15、以下程序的输出结果是_____。

```
int   f( ){
static int i=0;
int s=1;
s+=i;   i++;
return s;
}
main()
{ int i,a=0;
for(i=0;i<5;i++) a+=f( );
printf("%d\n",a);
}
```

A：20
B：24
C：25
D：15

16、在说明语句：int *f();中，标识符f代表的是_____。

A：一个用于指向整型数据的指针变量
B：一个用于指向一维数组的行指针
C：一个用于指向函数的指针变量
D：一个返回值为指针型的函数名

17、设有定义语句
int x[6]={2,4,6,8,5,7},*p=x,i;
要求依次输出x数组6个元素中的值，不能完成此操作的语句是_____。
A：for(i=0;i<6;i++) printf("%2d",*(p++));
B：for(i=0;i<6;i++) printf("%2d",*(p+i));
C：for(i=0;i<6;i++) printf("%2d",*p++);
D：for(i=0;i<6;i++) printf("%2d",(*p)++);

18、以下叙述正确的是_____。
A：C语言比其他语言高级
B：C语言可以不用编译就能被计算机识别执行
C：C语言以接近英语国家的自然语言和数学语言作为语言的表达形式
D：C语言比其他高级语言出现的最晚、具有其他语言的一切优点

19、以下叙述中不正确的是_____。
A：C语言中的文本文件以ASCII码形式存储数据
B：C语言中对二进制位的访问速度比文本文件快
C：C语言中，随机读写方式不使用于文本文件
D：C语言中，顺序读写方式不使用于二进制文件

20、程序中头文件type1.h的内容是：
#define N 5
#define M1 N*3
程序如下：
#include "type1.h"
#define M2 N*2
main()
{ int i;
i=M1+M2; printf("%d\n",i);
}
程序编译后运行的输出结果是_____。

A：10
B：20
C：25
D：30

C 语言机试测试卷（四）解析

1、答案：B
解析：本题中主函数里的x、y，fun函数里的a，b，k，这些都是指针，fun函数中只是将a，b这两个指针交换了位置，而并没有改变主函数中变量a，b的值。
本题答案为B。

2、答案：A

解析：本题的考查点是二维数组和指针数组的应用。
在所给的程序中，pa指向二维数组首地址a，接着执行下面的for循环，当i=0或i=1时，数组元素 pa[1][i]的值减1，当 i=2时，数组元素pa[1][i]的值为1，此时得到新的二维数组a[][3]={{1,2,3},{3,4,1}}，最终打印结果就是2+4+1=7。
故本题答案为A。

3、答案：B
解析：此题考的是指向数组元素的指针变量。
本题首先要知道，C语言中二维数组元素排列的顺序是按行存放，即在内存中先顺序存放第一行的元素，再存放第二行的元素。此题二维数组str[0]前5个元素为"C'，'h'，'i'，'n'，'a'，'\0'，第6个元素为'\0'，后4个元素为空字符。紧接着内存单元存放的内容即从str[1][0]到str[1][9]为'B'，'e'，'i'，'j'，'i'，'n'，'g'，'\0'，第八个元素为'\0'，后两个元素为空字符，执行*p=str后，指针p就指向了二维数组str的首地址，所以p+10就指向了从首地址依次向后的第11个存储单元即字符B的地址，所以输出Beijing，故答案为B。如果此题将*p=str改为(*p)[10]=str;则p+10指的就是str的第10行，如果此时要想输出Beijing，则必需将p+10改为p+1，因为(*p)[10]中的p是行指针，它是指向包含10个元素的一维数组，那么p+1就是加的一行了；而本题的p+1只是移动一个位置，因为此题的p只是一个指针变量。
故本题答案为B。

4、答案：D
解析：本题考查的是数组元素的引用。
如果定义一个指针指向二维数组的一行，则可以移动到下一行，这就是行指针。行指针的定义格式为：[存储类型] 数据类型(*指针)[n] 这里的n是一个整数，表示所指向的二维数组的列数。(*p)[5]中p是指向一个包含5个元素的一维数组，p的值就是该一维数组的首地址。所以当用行指针访问二维数组时，行指针每增加1，就移动一行，即指向二维数组的下一行，例如此题中的p+i将指向数组的第i行。即选项A所指的意思是指向数组c的第1行，并不是引用c数组的元素；B答案中p+3是指向数组元素的第3行，*(p+3)是指第三行第零个元素的地址；选项C由对选项A，选项B的解释可知，它指的是第一行第3列元素的地址；选项D中p[0]+2是指第0行第2列元素的地址，再加个*就表示第0行第2列的元素，即c[0][2]。
故本题答案为D。

5、答案：D
解析：本题的考查点是强制类型转换。
不论p是指向什么类型的指针变量，都可以通过强制类型转换的方法使之类型一致，强制类型转换的格式为（数据类型 *）。
故本题答案为D。

6、答案：D
解析：本题的考查点是数组元素的引用。
cp=c这个语句是将数组c的首地址赋给了指针数组cp的第一个数组元素。
选项A，cp+1是指将数组c的首地址加1，不是对数组元素的引用；
选项B，*(cp+3)是地址等于数组c的首地址加3的那个内存单元的内容，不是对数组元素的引用；

选项C，*(cp+1)+3是地址等于数组c的首地址加1的那个内存单元中存放的值加3，不是对数组元素的引用。
故本题答案为D。

7、答案：B
解析：在C语言中，宏定义在编译时将被直接替换，所以NUM最后会被替换成2*N+1+1，即2*2+1+1，值为6。因此，for循环执行的次数为6。
本题答案为B。

8、答案：A
解析：本题考查的是排序算法。
for(i=0;i<3;i++)
for (j=2;j>=i;j--)
if(x[j+1]>x[j]){ k=x[j];x[j]=x[j+1];x[j+1]=k;}
此段程序的作用是将数组x[0]到x[3]中的数由大到小进行排列，运行此段程序后x[]中的值变为x[]={7,5,3,1,2,4,6,0}
for(i=0;i<3;i++)
for(j=4;j<7-i;j++)
if(x[j]>x[j+1]){ k=x[j];x[j]=x[j+1];x[j+1]=k;}
此段程序的作用是将数组x[4]到x[7]中的数由小到大进行排列，运行此段程序后x[]中的值变为x[]={7,5,3,1,0,2,4,6}
最后通过一个循环将X[]中的数依次输出。
故本题答案为A。

9、答案：B
解析：本题所考查的是宏定义。C语言在预编译时遇到带实参的宏名，则按命令行中指定的字符串从左到右进行置换。在做这条题时，我们也不妨将运用置换法。
得到：a/=k+m*k+m/k+m*k+m=16/7=2。
注：a为整形，所以在做除法时，自动取整。
本题答案为B。

10、答案：C
解析：通过程序中给出的语句，我们可以判断，*p 是一个指向 double 型指针的指针。因此，要通过 p 把动态分配存储单元的地址传回主调函数，应该使用 double **p。本题答案为 C。

11、答案：D
解析：本题考查C语言数组的定义与初始化。
在C语言中数组的初始化可以对部分元素，数组的定义，不能含有变量。
本题答案为D。

12、答案：D
解析：本题主要考查C语言中构造类型，联合体与结构体。结构体中的成员使用各自的存储区，而联合体中的成员使用共同的存储区。
本题答案为D。

13、答案：A
解析：在scanf语句中，要为某一变量赋值，引用的是该变量的地址，所以在这里要为a,b,c 赋值可以写成两种形式，除了采用答案B的形式外，还可以写成scanf("%d%d%d",&a,&b,d)。
本题答案为A。

14、答案：A
解析：switch结构的执行过程为：进入switch结构后，对条件表达式进行计算，然后从上至下去找与条件表达式的值相匹配的case,以此作为入口，执行switch结构中后面的各语句。第一次for循环中，switch结构的条件表达式i的值是1，则从case 1 开始执行后面的语句，先执行a+=3，a的值变成3，接着执行a+=5，a的值变成了8。然后进入第二次循环，switch结构条件表达式i的值2，则从case 2开始执行后面的语句，结束第二轮循环时a的值是16，在第三轮循环中，switch结构条件表达式的值是3，则从case 3后面的语句开始执行，a+=2 语句使a的值变成了18，然后执行a+=3，和a+=5，第三轮循环结束时，a的值是26。第四轮循环中， switch结构条件表达的值是4，从default处开始执行，执行一个a+=5的操作，a的值变成了31，i 的值经修改变成了5，不再满足循环条件，退出循环。
本题答案为A。

15、答案：D
解析：本题主要考的是对变量存储属性的理解，一个变量被指定为静态变量，在编译时就为其分配了存储空间，程序一开始执行便被建立，直到该程序执行结束都存在，而不像动态变量只存在于函数或分程序被调用期间。在函数多次被调用的过程中静态局部变量的值具有可继承性。在第一次调用函数结束时，i的值是1，返回值1，第二次调用函数时，i的值保持为1，执行自加操作后，i的值变成了2，到了第三次调用时，i的值保持了上次调用结束时的值，再执行自加，值变成了3。到第五次调用结束时，i的值是5。而s的值不具备i的这种可继承性，每次调用时，它都先被重新赋值为1，再执行下面的操作。
本题答案为D。

16、答案：D
解析：int * f() 表示f是一个函数，它带回一个指针值，这个指针是指向一个整型数据的。
本题答案为D。

17、答案：D
解析：此题考的是指向数组元素的指针。
p=x的作用是将x的首地址（即x[0]的地址）赋给指针变量p，那么程序段A当i=0时，(p++)虽然是先执行括号内的表达式，但括号内表达的++是在p的后面，所以它要先执行p再加，即先取*p(x[0])，再使得p加1,为i=2时取下一个元素作准备。选项C中因为*和++是处于同一优先级别，而结合方向为自右而左，因此*p++相当于*(p++)，所以选项A和选项C都是可以输出6个元素的值的。
选项B，p表示x的首地址即x[0]的地址，p+i则表示x[i]的地址，那么*(p+i)就表示x[i]的值，所以选项C也是可以输出6个元素的值的；选项D由于(*p)++后面的++

是使*p的值+1，又因为*p指的是x[0]的值，所以它不能实现地址的依次向下移动，每次循环只是x[0]的值在改变，而p的值并没有丝毫变化，所以它并不能实现6个元素的输出。
故本题答案为D。

18、答案：C
解析：本题的考查点是C语言的特点。
C语言主要有如下一些特点：
1，语言简洁、紧凑，使用方便、灵活；
2，运算符丰富；
3，数据结构丰富，具有现代化语言的各种数据结构；
4，具有结构化的控制语句；
5，语法限制不太严格，程序设计自由度大；
6，C语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。
7，生成目标代码质量高，程序执行效率高；
8，用C语言写的程序可移植性好。
但是所有的语言都不可能没有任何缺点，C语言也不例外，C语言是高级语言，需要通过编译才能被计算机识别。本题答案为C。

19、答案：D
解析：本题主要考查C文件的基本概念。
C语言把文件看作是一个字符（字节）的序列，根据数据组织形式可分为ASCII文件（又称文本文件）和二进制文件，即一个C文件可以看成是一个字节流或二进制流，对于流式文件，可以进行顺序读写，也可以进行随机读写，关键在于控制文件的位置指针，由于文本文件要发生字符转换，计算位置时往往会发生混乱，访问速度受到影响，所以随机读写方式不使用于文本文件。
故本题答案为D。

20、答案：C
解析：本题的考查点是宏定义。
宏定义就是用一个指定的标识符（即名字）来代表一个字符串，它的一般形式为：
#define　标识符　字符串
这种方法使用户能以一个简单的名字代替一个长的字符串，因此把这个标识符（名字）称为"宏名"。宏名一般习惯用大写字母表示，以与变量名相区别。使用宏名代替一个字符串，可以减少程序中重复书写某些字符串的工作量。
i=M1+M2=N*3+N*2=5*3+5*2=25。
故本题答案为 C。

C 语言机试测试卷（五）

1、若有语句：char　*line[5];，以下叙述中正确的是_____。
A：定义line是一个数组，每个数组元素是一个基类型为char的指针变量
B：定义line是一个指针变量，该变量可以指向一个长度为5的字符型数组
C：定义line是一个指针数组，语句中的*号称为间址运算符

D：定义line是一个指向字符型函数的指针

2、以下选项中，能定义s为合法的结构体变量的是_____。

A：typedef struct abc
{ double a;
char b[10];
} s;

B：struct
{ double a;
char b[10];
}s;

C：struct ABC
{ double a;
char b[10];
}
ABC s;

D：typedef ABC
{ double a;
char b[10];
}
ABC s;

3、若有下面的说明和定义，则sizeof(struct aa)的值是_____。

```
struct aa
{   int r1;double r2;float r3;
union uu{
char u1[5];
long u2[2];
}ua;
}maya;
```

A：30
B：29
C：24
D：22

4、以下各选项企图说明一种新的类型名，其中正确的是_____。

A：typedef v1 int;

B：typedef v2=int;
C：typedef int v3;
D：typedef v4: int

5、设有如下定义：
```
struct sk{
int a;
float b;
}data;
int *p;
```
若要使p指向data中的a域，正确的赋值语句是_____。
A：p=&a;
B：p=data.a;
C：p=&data.a;
D：*p=data.a

6、若要说明一个类型名STP，使得定义语句STP s;等价于char *s;，以下选项中正确的是_____。

A：typedef STP char *s;
B：typedef *char STP;
C：typedef STP *char;
D：typedef char* STP;

7、设有如下定义：
```
struct  ss
{   char  name[10];
int   age;
char  sex;
}std[3],*p=std;
```
下面各输入语句中错误的是_____。
A：scanf("%d",&(*p).age);
B：scanf("%s",&std.name);
C：scanf("%c",&std[0].sex);
D：scanf("%c",&(p->sex));

8、以下程序的功能是：建立一个带有头结点的单向链表，并将存储在数组中的字符依次转储到链表的各个结点中，请为下划线处有号码的选择出正确的选项。
```
#include <stdlib.h>
stuct node
{ char data; struct node *next;};
```

```
_____  CreatList(char *s)
{ struct node *h,*p,*q);
h=(struct node *) malloc(sizeof(struct node));
p=q=h;
while(*s!='\0')
{   p=(struct node *) malloc(sizeof(struct node));
p->data=___(49)___  ;
q->next=p;
q=_____  ;
s++;
}
p->next='\0';
return h;
}
main()
{   char str[]="link list";
struct node *head;
head=CreatList(str);
...
}
```

A：*s
B：s
C：*s++
D：(*s)++

9、若有以下说明和定义

```
union dt
{int a;char b;double c;} data;
```

以下叙述中错误的是_____。

A：data的每个成员起始地址都相同
B：变量data所占的内存字节数与成员c所占字节数相等
C：程序段：data.a=5;printf("%f\n",data.c);输出结果为5.000000
D：data可以作为函数的实参

10、有以下程序段

```
struct st{
int x;int *y;}*pt;
int a[]={1,2},b[]={3,4};
struct st c[2]={10,a,20,b};
pt=c;
```

以下选项中表达式的值为11的是_____。

A：*pt->y
B：pt->x
C：++pt->x
D：(pt++)->x

11、下面描述中，符合结构化程序设计风格的是_____。

A：使用顺序、选择和重复（循环）三种基本控制结构表示程序的控制逻辑
B：模块只有一个入口，可以有多个出口
C：注重提高程序的执行效率
D：不使用goto语句

12、结构化程序设计主要强调的是_____。
A：程序的规模
B：程序的易读性
C：程序的执行效率
D：程序的可移植性

13、有以下程序

```
main() {
int a=1,b;
for(b=1;b<=10;b++)
{   if(a>=8) break;
if(a%2==1){a+=5;continue;}
a-=3;
}
printf("%d\n",b);
}
```

程序运行后的输出结果是_____。
A：3
B：4
C：5
D：6

14、在嵌套使用if语句时，C语言规定else总是_____。
A：和之前与其具有相同缩进位置的if配对
B：和之前与其最近的if配对
C：和之前与其最近的且不带else的if配对
D：和之前的第一个if配对

15、若a,b,c1,c2,x,y均是整型变量，正确的switch语句是_____。

①

```
switch(a+b);
{   case 1:y=a+b;break;
case 0:y=a-b;break;
}
```

②

```
switch(a*a+b*b)
{   case 3:
case 1:y=a+b;break;
case 3:y=b-a;break;
}
```

③

```
switch  a
{   case c1:y=a-b;break;
case c2:x=a*b;break;
default:x=a+b;
}
```

④

```
switch (a-b)
{   default:y=a*b;break;
case  3:case 4:x=a+b;break;
case 10:case 11:y=a-b;break;
}
```

A：①
B：②
C：③
D：④

16、C语言中用于结构化程序设计的三种基本结构是_____。

A：顺序结构、选择结构、循环结构
B：if、switch、break
C：for、while、do-while
D：if、for、continue

17、以下程序中，while循环的循环次数是_____。

```
main()
{int i=0;
```

```
while(i<10)
{  if(i<1) continue;
if(i==5)break;
i++;
}
......
}
```

A：1
B：10
C：6
D：死循环，不能确定次数

18、以下程序的功能是：按顺序读入10名学生4门课程的成绩，计算出每位学生的平均分并输出，程序如下：

```
main(){
int n,k;
float score,sum,ave;
sum=0.0;
for(n=1;n<=10;n++)
{for(k=1;k<=4;k++)
{scanf("%f",&score);  sum+=score;}
ave=sum/4.0;
printf("NO%d:%f\n",n,ave);
}
}
```

上述程序运行后结果不正确，调试中发现有一条语句出现在程序的位置不正确。这条语句是_____。

A：sum=0.0;
B：sum+=score;
C：ave=sum/4.0;
D：printf("NO%d:%f\n",n,ave);

19、有以下程序段

```
int n=0,p;
do {scanf("%d",&p);n++;}  while(p!=12345&&n<3);
```

此处do-while循环的结束条件是_____。

A：p的值不等于12345并且n的值小于3
B：p的值等于12345并且n的值大于等于3
C：p的值不等于12345或者n的值小于3
D：p的值等于12345或者n的值大于等于3

20、结构化程序由三种基本结构组成，三种基本结构组成的算法_____。
A：可以完成任何复杂的任务
B：只能完成部分复杂的任务
C：只能完成符合结构化的任务
D：只能完成一些简单的任务

C 语言机试测试卷（五）解析

1、答案：A
解析：本题考查的是指针与数组。
由于运算符[ ]优先级比*高，所以"char *line[5];"相当于"char *(line[5]);"，表示line是一个数组，每个数组元素是一个基类型为char的指针变量。
故本题答案为A。

2、答案：B
解析：本题的考查点是结构体变量的定义。
定义一个结构体类型的变量，可采用三种方法：
(1)先定义结构体类型再定义变量名；
(2)在定义类型的同时定义变量；
(3)直接定义结构类型变量，即不出现结构体名；
选项B符合第三种定义方法。
故本题答案为B。

3、答案：D
解析：本题的考查点是结构体的长度。
sizeof(x)为运算符，它运算的结果是x型的数据结构占用的内存字节数，题目中定义了一个结构体aa，它的长度为其中数据成员所占内存的总和。
int型的变量需要2个字节；double型的变量需要8个字节；float型的变量需要4个字节；题目中结构体aa包含了一个共用体uu，而共用体变量在内存中所占的长度等于最长的成员的长度，而char型的变量需要1个字节，char u1[5]即需要5个字节，long型的变量需要4个字节，long u2[2]即需要8个字节，所以共用体uu需要8个字节的存储空间；因此，sizeof(struct aa)的值应为2+8+4+8=22个字节。
故本题答案为D。

4、答案：C
解析：本题考查C语言的类型定义。
C语言中可以使用typedef来重新定义已有的数据类型，相当于为数据类型取个别名。
本题答案为C。

5、答案：C
解析：本题的考查点是指向结构体变量的指针。
将data.a的起始地址赋给指针变量p，也就是使p指向data.a。

故本题答案为C。

6、答案：D
解析：本题的考查点是类型定义typedef。
C语言用类型定义把已有的类型标识符定义成新的类型标识符，经类型定义后，新的类型标识符即可当做原标识符使用。它的一般形式为：
typedef　原类型标识符　新类型标识符
定义一个新的类型名的方法是：
1,先按定义变量(对字符指针类型方法相同)的方法写出定义体(如:char　*s;);
2，将变量名换成新类型名（如：将s换成STP）；
3，在最前面加上typedef（如：typedef char *STP;）。
故本题答案为D。

7、答案：B
解析：本题的考查点是结构体数组的引用。
在C语言中，结构体变量的引用有三种等价方式：
1，结构体变量.成员名；
2，（*p）.成员名；
3，p->成员名。
其中（*p）表示p指向的结构体变量。结构体数组的引用方法和结构体变量的引用类似，根据格式输入函数scanf()的特点，地址表列是由若干个地址组成的表列。题目中的 *p=std;表示p指向结构体数组std，选项A，D正确；C语言中规定第一个元素的地址即为该数组的首地址，所以选项C也正确。
故本题答案为B。

8、答案：A
解析：本题考的是建立动态链表。
从本题要求建立一个struct node类型的数据链表，从main()函数中可知，head是"头指针"变量，它的值是由CreatList(str)带回的一个结构体类型的指针变量，指向所建立的表的第一个数据，所以第48空应该填struct node *。再从本题的函数参数传递看，传递的是字符串link list的首地址给指针变量s，所以可以推断建立的链表一定与link list字符串有关，再看CreatList(char *s)函数中所定义的变量及其它语句，可知h,p,q 是用与建立链表的，h表示头指针，p用于记录开辟的新结点，而q是用于将新结点与已建立链表相连的中间变量，且所建立链表各个结点的data依次存放的是link list字符串的各个字符，所以49空填的是*s。
故本题答案为A。

9、答案：C
解析：本题考查的是共用体。
union是表示共用体的关键字，说明其内的成员a,b,c共占用同一个内存空间，所以data的每个成员起始地址都相同;共用体变量所占的内存长度等于最长的成员的长度，故变量data所占的内存字节数与成员c所占字节数相等；但由于各成员变量的类型不一致，所以它们的存储方式也不相同，整型占用2个字节的存储空

间，double型占用8个字节的存储空间；另外data可以作为函数的实参，此时data作为一个地址进行地址传送；执行"data.a=5;printf("%f\n",data.c);"系统不能自动将整型变量转换为浮点型值,printf函数只是将内存中的数据以不同的类型输出,而不能将内存中的整型数据自动转换为等值的浮点数,故C是错误的。
故本题答案为C。

10、答案：C
解析：本题考查的是结构体类型的指针。
题目中定义了一个st结构体类型，然后定义了st型的结构体指针变量*pt及结构体数组c，并对结构体数组进行了初始化，且将结构体指针变量pt指向了数组c，则pt->x的值为10，则++pt->x的值为11（注："->"运算符的优先级高于"++"），故选项C正确。
故本题答案为C。

11、答案：A
解析：结构化程序设计方法的四条原则是：1.自顶向下；2.逐步求精；3.模块化；4.限制使用goto语句。
"自顶向下"是说，程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标；"逐步求精"是说，对复杂问题，应设计一些子目标，作过渡，逐步细节化；"模块化"是说，一个复杂问题，肯定是由若干稍简单的问题构成；解决这个复杂问题的程序，也应对应若干稍简单的问题，分解成若干稍小的部分。
本题答案为A。

12、答案：B
解析：结构化程序设计主要强调的是结构化程序清晰易读，可理解性好，程序员能够进行逐步求精、程序证明和测试，以保证程序的正确性。
本题答案为B。

13、答案：B
解析：本题考查的是break与continue的区别。
break和continue的区别是：continue语句只结束本次循环，而不是终止整个循环的执行；而break语句则是结束整个循环过程，不再判断执行的条件是否成立。
本题具体执行过程如下：
a=1,b=1：a>=8不成立，判断a%2==1成立，a+=5则a=6，continue，执行下一次循环；
a=6,b=2：a>=8不成立，判断a%2==1不成立，a-=3则a=3，执行下一次循环；
a=3,b=3：a>=8不成立，判断a%2==1成立，a+=5则a=8，continue，执行下一次循环；
a=8,b=4：a>=8成立，break，结束整个循环，输出b的值4。
故本题答案为B。

14、答案：C
解析：本题考查的是IF语句的使用规则。
在嵌套使用if语句时，C语言规定else总是和之前与其最近的且不带else的if配

对。故本题答案为C。

15、答案：D
解析：选项A的错误在于switch语句后面不该加分号；
选项B的错误在于switch语句要求每一个case的常量表达式的值必须互不相同；
选项C的错误在于变量a应该用括号括起来，case后面应该是常量表达式，而不是变量。
本题答案为D。

16、答案：A
解析：本题的考查点是程序的基本结构。
结构化程序由若干个基本结构组成。每一个基本结构可以包含一个或若干个语句。有三种基本结构：
1.顺序结构，先执行A操作，再执行B操作，两者是顺序执行的关系；
2.选择结构，P代表一个条件，当P条件成立时执行A，否则执行B。只能执行A或B之一。
3.循环结构，有两种循环结构：
（1）当型循环结构，当P条件成立（"真"）时，反复执行A操作。直到P为"假"时才停止循环。
（2）直到型循环结构，先执行A操作，再判断P是否为"假"，若P为"假"，再执行A，如此反复，直到P为"真"为止。
故本题答案为A。

17、答案：D
解析：题目中，进入循环后，先执行一个条件语句，如果i的值小于1，那么直接进入下一轮循环，因为i的初始值是0，小于1，故直接进入下一轮循环，又因为i的值始终没有改变，所以这个循环成了死循环。
本题答案为D。

18、答案：A
解析：本题的考查点是for循环语句的使用。
题目中要求每位学生的平均分数，所以要求每位学生的总分，也就是程序中的sum，所以sum应在for循环体内赋初值，如果在循环体外赋初值就会得到所有学生的分数总和，不合题意。
故本题答案为A。

19、答案：D
解析：本题的考查点是do-while循环语句。
当循环条件为假时跳出循环，因为条件为逻辑与，所以只要有一个不满足条件即跳出循环。
故本题答案为D。

20、答案：C
解析：本题考查的是结构化程序的任务。

C 语言的算法是由三种基本结构（顺序结构、选择结构、循环结构）组成的，用三种基本结构组成的程序必然是结构化的程序，结构化的程序完成的是符合结构化的任务。故本题答案为 C。

C 语言机试测试卷（六）

1、有以下程序

```
main()
{ int x=0,y=5,z=3;
while(z-->0&&++x<5) y=y-1;
printf("%d,%d,%d\n",x,y,z);
}
```

程序执行后的输出结果是_____。

A：3,2,0
B：3,2,-1
C：4,3,-1
D：5,-2,-5

2、以下4个选项，不能看作一条语句的是_____。

A：{;}
B：a=0,b=0,c=0;
C：if(a>0);
D：if(b==0)m=1;n=2;

3、有以下程序

```
main()
{ int a=1,b=2,m=0,n=0,k;
k=(n=b>a)||(m=a<b);
printf("%d,%d\n",k,m);
}
```

程序运行后的输出结果是_____。

A：0,0
B：0,1
C：1,0
D：1,1

4、有以下程序

```
main()
{ int  i=0,s=0;
do{
if(i%2){i++;continue;}
i++;
```

```
s+=i;
} while(i<7);
printf("%d\n",s);
}
```
执行后输出的结果是_____。

A：16
B：12
C：28
D：21

5、有以下程序
```
main()
{ int a=5,b=4,c=3,d=2;
if(a>b>c)
printf("%d\n",d);
else if((c-1>=d)==1)
printf("%d\n",d+1);
else
printf("%d\n",d+2);
}
```
执行后输出的结果是_____。

A：2
B：3
C：4
D：编译时有错，无结果

6、设有以下语句
```
char str[4][12]={"aaa","bbbb","ccccc","dddddd"},*strp[4];
int i;
for(i=0;i<4;i++)strp[i]=str[i];
```
下列选择不是对字符正确引用的是(其中0≤k<4)_____。

A：strp
B：str[k]
C：strp[k]
D：*strp

7、若有以下程序
```
#include   <stdio.h>
void f(int n);
main()
```

```
{   void f(int n);
f(5);
}
void f(int n)
{ printf("%d\n",n); }
```
则以下叙述中不正确的是_____。

A：若只在主函数中对函数f进行说明，则只能在主函数中正确调用函数f
B：若在主函数前对函数f进行说明，则在主函数和其后的其它函数中都可以正确调用函数f
C：对于以上函数程序，编译时系统会提示出错信息；提示对f函数重复说明
D：函数f无返回值，所以可用void将其类型定义为无值型

8、有以下程序

```
void  fun(int *a,int i,int  j)
{   int  t;
if (i<j)
{   t=a[i];a[i]=a[j];a[j]=t;
fun(a,++i,--j);
}
}
main()
{ int  a[]={1,2,3,4,5,6},i;
fun(a,0,5);
for(i=0;i<6;i++)
printf("%d ",a[i]);
}
```
执行后输出结果是_____。

A：6 5 4 3 2 1
B：4 3 2 1 5 6
C：4 5 6 1 2 3
D：1 2 3 4 5 6

9、若已定义的函数有返回值，则以下关于该函数调用的叙述中错误的是_____。

A：函数调用可以作为独立的语句存在
B：函数调用可以作为一个函数的实参
C：函数调用可以出现在表达式中
D：函数调用可以作为一个函数的形参

10、下列函数定义中，会出现编译错误的是_____。

A：
```
max(int x,int y,int *z)
{*z=x>y ? x:y;}
```

B：int max(int x,y)
{ int z;
z=x>y ? x:y;
return z;}

C：max(int x,int y)
{ int z;
z=x>y?x:y; return(z);}

D：int max(int x,int y)
{ return(x>y?x:y); }

11、有以下程序
```
void f(int a[],int i, int j)
{   int t;
if(i<j)
{   t=a[i];a[i]=a[j];a[j]=t;
f(a,i+1,j-1);
}
}
main()
{   int i,aa[5]={1,2,3,4,5};
f(aa,0,4);
for(i=0;i<5;i++)printf("%d,",aa[i]);printf("\n");
}
```
执行后输出结果是_____。

A：5,4,3,2,1,
B：5,2,3,4,1,
C：1,2,3,4,5,
D：1,5,4,3,2,

12、有以下程序
```
int f(int a)
{ return a%2; }
main()
{   int s[8]={1,3,5,2,4,6},i,d=0;
for (i=0;f(s[i]);i++) d+=s[i];
printf("%d\n",d);
}
```
程序运行后的输出结果是_____。

A：9

B：11
C：19
D：21

13、在函数调用过程中，如果函数funA调用了函数 funB，函数funB又调用了函数funA，则_____。

A：称为函数的直接递归调用
B：称为函数的间接递归调用
C：称为函数的循环调用
D：C语言中不允许这样的递归调用

14、在一个C语言程序中_____。
A：main函数必须出现在所有函数之前
B：main函数可以在任何地方出现
C：main函数必须出现在所有函数之后
D：main函数必须出现在固定位置

15、下列叙述中正确的是_____。
A：C语言中既有逻辑类型也有集合类型
B：C语言中没有逻辑类型但有集合类型
C：C语言中有逻辑类型但没有集合类型
D：C语言中既没有逻辑类型也没有集合类型

16、下列关于单目运算符++、--的叙述中正确的是_____。
A：它们的运算对象可以是任何变量和常量
B：它们的运算对象可以是char型变量和int型变量，但不能是float型变量
C：它们的运算对象可以是int型变量，但不能是double型变量和float型变量
D：它们的运算对象可以是char型变量、int型变量和float型变量

17、若有以下程序段：
```
int  m=0xabc,n=0xabc;
m-=n;
printf("%X\n",m);
```
执行后输出结果是_____。

A：0X0
B：0x0
C：0
D：0XABC

18、以下选项中，合法的一组C语言数值常量是_____。
A：028　　.5e-3　　-0xf

B：12.　　0xa23　　4.5e0
C：.177　　4e1.5　　0abc
D：0x8A　　10,000　　3.e5

19、以下关于long、int和short类型数据占用内存大小的叙述中正确的是_____。
A：均占4个字节
B：根据数据的大小来决定所占内存的字节数
C：由用户自己定义
D：由C语言编译系统决定

20、以下程序的功能是进行位运算
```
main()
{   unsigned char  a,b;
a=7^3;  b=~4&3;
printf("%d  %d\n",a,b);
}
```
程序运行后的输出结果是_____。
A：4　3
B：7　3
C：7　0
D：4　0

C 语言机试测试卷（六）解析

1、答案：B
解析：当x=0,y=5,z=3时，z-->0&&++x<5成立，执行y=y-1;此时，y=5-1=4,z=z-1=2,x=x+1=1。执行后条件仍然成立，继续执行y=y-1，此时y=y-1=3,z=1,x=2，以此类推，直至条件不成立。当z为0时条件不成立，执行后z为-1。
故本题答案为B。

2、答案：D
解析：if语句是用来判定所给的条件是否满足，根据判定的结果（真或假）决定执行给出的两种操作之一。
在if和else后面可以只含一个内嵌的操作语句，也可以有多个操作语句，此时用花括号"{}"将几个语句括起来成为一个复合语句。选项D中没有将两个操作语句括起来，不能看作一条语句。
故本题答案为D。

3、答案：C
解析：本题的考查点是逻辑或运算。
a||b　若a,b之一为真，则a||b为真，但如果a为真，则b不用考虑，也就是无需计算，仍为原值。
结合本题，因为b>a为真，即n=1，所以m不用计算仍为原值，且k值为1。

故本题答案为C。

4、答案：A
解析：本题的考查点是do-while循环语句。
do-while的一般格式如下所示：
do
{语句;
}while(测试表达式);
do-while语句是先执行后判断，其执行流程是：
1. 执行循环体；
2. 计算测试表达式，若为非0，转向执行1，否则执行3；
3. 执行do-while后面的语句，即退出do-while循环。
本题中，当i=0时，i%2=0，为假不执行{i++;continue;}，接着往下执行i++，此时i=1，s=s+i=1，因为i=1符合条件i<7继续循环执行，i=1时以此类推，直到i>=7时结束循环。最后得出s=16。
故本题答案为A。

5、答案：B
解析：本题的考查点是关系表达式和逻辑表达式。
关系表达式用来进行两个数据的比较，比较的结果为逻辑"真"或逻辑"假"。C语言不提供逻辑型数据，而是用整数0表示逻辑"假"，即比较不成立；用整数1表示逻辑"真"，即比较成立。
和关系表达式一样，逻辑表达式的值也是用整数1表示逻辑"真"，用整数0表示逻辑"假"。
本题中'a>b>c'即相当于'(a>b)>c'因为a>b成立，所以其值为1，然后比较1与c的大小，1>c不成立，所以不执行printf语句，然后判断'(c-1>=d)==1'成立，此时执行printf语句，打印d+1=3。
故本题答案为B。

6、答案：A
解析：本题的考查点是对字符数组的引用。
单独的一个"strp"代表的是指针数组strp的第一个数组元素的地址，不是对字符的引用。
故本题答案为A。

7、答案：C
解析：一个函数在一个文件中的定义只能有一次，但对它的声明却可以有很多个。
本题答案为C。

8、答案：A
解析：本题的考查点是函数的递归调用。
所谓函数的递归调用就是在调用一个函数的过程中又直接或间接地调用该函数本身。程序在执行被调函数的过程中，先判断i<j为真，交换a[i]和a[j]，再次调用函数本身，i自加1，j自减1，判断出i<j为真，交换a[i]和a[j]，……，最

终打印输出6 5 4 3 2 1。
故本题答案为A。

9、答案：D
解析：本题的考查点是函数的调用。
在调用函数时，大多数情况下，主调函数和被调函数之间有数据传递关系。这就是平时所说的实参和形参，在函数调用时，函数名后面括弧里的表达式就称为"实参"。
按函数在程序中出现的位置来分，可以有以下三种函数调用形式：
1.把函数调用作为一个语句。如：printstar(); 这时不要求函数带回值，只要求函数完成一定的操作。故A选项的说法是对的。
2.函数出现在表达式中，这种表达式称为函数表达式。这时要求函数带回一个确定的值以参加表达式的运算。例如： c=2*max(a,b) 函数max是表达式的一部分，它的值乘2再赋给c。所以选项C的说法是正确的。
3.函数调用作为一个函数的实参，例：m=max(a,max(b,c)); 其中max(b,c)是一次函数调用，它的值作为max另一次调用的实参。但函数调用不可以作为一个函数的形参，因为函数调用的参数的数据传递是"值传递"，即单向传递，只由实参传给形参，不能由形参传回来给实参。故B选项的说法也是正确的。选项D错误。
故本题答案为D。

10、答案：B
解析：本题的考查点是函数的返回值。
函数的返回值是由return语句带回的，如果被调用的函数中没有return语句，并不带回一个确定的、用户所希望得到的函数值，但实际上，函数并不是不带回值，而只是不带回有用的值，带回的是一个不确定的值。但并不影响程序的运行。故选项A,C 虽然没有return语句，但没有语法错误，而选项B中，定义max函数时，里面的形参也要分别定义。而选项中只定义了x的数据类型，而y并未定义，所以会出现错误。
故本题答案为B。

11、答案：A
解析：f函数的的功能是通过递归调用实现数组中左右部分相应位置数据的交换。即数组中第一个元素与最后一个调换位置，第二个与倒第二个调换位置，以此类推……。
故本题答案为A。

12、答案：A
解析：此题考的是for语句的循环终止条件。
在主函数main()中，f(s[i])为for语句的循环终止条件，当f函数返回的值非零时，则继续执行循环，否则循环终止，输出d的值。下面是此题的运行步骤：当i=0时，判断f(s[i])所返回的值是否符合终止条件，此时返回值为1，符合条件，继续执行，d=1；当i=1时，s[1]=3传递给a,a%2=1，所以f()的返回值为1，符合条件，d=d+s[1]=4;当i=2时，s[2]=5传递给a,a%2=1，所以f()的返回值为1，符合条件，d=d+s[2]=9；当i=3时，s[3]=2传递给a,a%2=0，所以f()的返回值为0，

条件为假，所以终止循环，直接执行循环下面的语句printf()，输出d为9。
故本题答案选A。

13、答案：B
解析：本题考的是函数的递归调用。
在调用一个函数的过程中又出现直接或间接地调用该函数本身，称为函数的递归调用；在调用f1函数过程中要调用f2函数，而在调用f2函数过程中又要调用f1函数是间接调用，所以本题是函数的间接递归调用。
故本题答案为B。

14、答案：B
解析：本题的考查点是main函数。
一个C程序中必须有且只能有一个用"main"命名的主函数，其它函数由用户自行命名。main函数可以在任何地方出现。故本题答案为B。

15、答案：D
解析：本题的考查点是C语言中的基本数据类型。
数据类型是所允许的数据及其操作的集合，是高级语言的重要特征。程序设计中可以利用数据类型发现程序中的某些错误。
C语言提供了三大类数据类型，即基本类型、复合类型和地址类型
1,基本类型只代表单个数据；
2,复合类型由基本类型组合而成，可代表一批数据；
3,地址类型可直接表示内存中的地址。
C语言支持的五种基本数据类型：
字符型：表示单个字符；
整型：表示整数，包括基本整型、短整型、长整型和无符号整型；
浮点型：表示实数，精度为6-7位有效数字；
双精度型：表示实数，精度为15-16位有效数字；
无值类型：表示无返回值的函数或无定向指针。
故本题答案为D。

16、答案：D
解析：本题的考查点是自增、自减运算符。
自增、自减运算符的作用是使变量的值增1或减1，只能用于变量，包括char型变量、int型变量和float型变量，而不能用于常量或表达式。故本题答案为D。

17、答案：C
解析：
本题的考查点是不同类型数据的输入输出。
printf()函数的一般格式如下所示：
printf("格式控制字符串",输出项清单)
格式控制字符串中包括：
1.格式转换说明符：以"%"开头后跟一个格式字符，它用于输出数据格式的转换与控制；

2.转义字符：以"\"开头后跟一个字符，用于输出某些特殊意义的字符和不可显示字符；
3.其他字符：用于照原样显示的字符。printf()格式转换说明符："%X"按实际位数输出十六进制整数。本题因为m，n都定义为十六进制整数，且m-=n;的意思是将m与n的差值重新赋给m，所以m为0。故本题答案为C。

18、答案：B
解析：本题的考查点是C语言数值常量。
数值常量分为不同的类型：整型常量、实型常量、字符常量。整型常量就是整常数。在C语言中，使用的整常数有八进制、十六进制和十进制三种。十进制整常数：十进制整常数没有前缀，其数码取值为0～9；八进制整常数：八进制整常数必须以0开头，即以0作为八进制数的前缀，数码取值为0～7，八进制数通常是无符号数；十六进制整常数：十六进制整常数的前缀为0X或0x，其数码取值为0～9，A～F或a～f。
实型常量也称为实数或者浮点数。在C语言中，实数只采用十进制。它有二种形式：十进制小数形式、指数形式。十进制小数形式：由数码0～9和小数点组成；指数形式：由十进制数加阶码标志"e"或"E"以及阶码（阶码只能为整数，可以带符号）组成。其一般形式为：a E n（a为十进制数，n为十进制整数）。
本题中由于以0开头的常量为八进制，且八进制常量的数码取值为0～7，所以选项A中"028"与选项C中"0abc"的表示均不正确；由于指数形式的实型常量的阶码只能为整数，所以选项C中"4e1.5"的表示不正确；由于十进制整常数的数码取值为0～9，所以选项D中"10,000"的表示不正确。故本题答案为B。

19、答案：D
解析：本题的考查点是C语言数据类型所占内存。
C语言标准只规定了char类型占用内存大小为1，而long、int、short等类型数据占用内存大小由编译系统决定。如在TC编译环境下，long、int和short类型数据占用内存大小分别为4,2和2，而在VC++的编译环境下，long、int和short类型数据占用内存大小分别为4,4和2。故本题答案为D。

20、答案：A
解析：本题考查的是位运算。
C语言提供六种位运算符，按优先级由高到低的顺序分别为：取反（～）→左移（<<）和右移（>>）→按位与（&）→按位异或（∧）→按位或（|）。"∧"是按位异或运算，当对应位上的二进制数值相同时，异或的结果为0，对应位上的二进制数值不同时，异或的结果为1，本题"7∧3"相当于"0111∧0011"，所以a的值为4；"&"是按位与运算符，只有当所有对应位上的数值都为1时，结果才为1；"~"是按位取反。题中"~4&3" 相当于1011&0011=0011，所以b的值为3。
故本题答案为A。

C 语言机试测试卷（七）

1、以下叙述中正确的是_____。
A：预处理命令行必须位于源文件的开头

B：在源文件的一行上可以有多条预处理命令
C：宏名必须用大写字母表示
D：宏替换不占用程序的运行时间

2、以下叙述中正确的是_____。
A：预处理命令行必须位于C源程序的起始位置
B：在C语言中，预处理命令行都以"#"开头
C：每个C程序必须在开头包含预处理命令行：#include<stdio.h>
D：C语言的预处理不能实现宏定义和条件编译的功能

3、请读程序片段(字符串内没有空格字符)：
printf("%d\n",strlen("ATS\n012\1\\"));
上面程序片段的输出结果是_____。
A：11
B：10
C：9
D：8

4、请选出可用做C语言用户标识符的一组标识符_____。

①	②	③	④
void	a3_b3	For	2a
define	_123	_abc	DO
WORD	IF	case	sizeof

A：①
B：②
C：③
D：④

5、以下选项中可作为C语言合法常量的是_____。
A：-80
B：-080
C：-8e1.0
D：-80.0e

6、若整型变量a,b,c,d中的值依次为：1,4,3,2
则条件表达式a<b?a:c<d?c:d的值是_____。

A：1
B：2
C：3
D：4

7、设有定义：int a; float b; 执行scanf("%2d%f",&a,&b)；语句时，若

从键盘输入876　543.0<回车>，a和b的值分别是_____。

A：876和543.000000
B：87和6.000000
C：87和543.000000
D：76和543.000000

8、若x和y都是int型变量，x=100,y=200，且有下面的程序片段
printf("%d",(x,y));
上面程序片段的输出结果是_____。

A：200
B：100
C：100　200
D：输入格式符不够，输出不确定的值

9、设a,b和c都是int型变量，且a=3,b=4,c=5，则下面的表达式中，值为0的表达式是_____。

A：'a'&&'b'
B：a<=b
C：a||+c&&b-c
D：!((a<b)&&!c||1)

10、在C语言中，合法的长整型常数是_____。

A：0L
B：4962710
C：324562&
D：216D

11、有以下程序
main()
{　int　x=102,y=012;
printf("%2d,%2d\n",x,y);
}
执行后输出结果是_____。
A：10,01
B：02,12
C：102,10
D：02,10

12、设有说明int(*ptr)[M];，其中的标识符ptr是_____。

A：M个指向整型变量的指针
B：指向M个整型变量的函数指针
C：一个指向具有M个整型元素的一维数组的指针
D：具有M个指针元素的一维指针数组，每个元素都只能指向整型变量

13、设有以下语句，其中对数组元素a不正确的引用是：_____(其中0≤i<10)
```
int a[10]={0,1,2,3,4,5,6,7,8,9,},*p=a;
```
A：a[p-a]
B：*(&a[i])
C：p[i]
D：*(*(a+i))

14、下面程序
```
int aa[3][3]={{2},{4},{6}};
main()
{   int i,*p=&aa[0][0];
for (i=0;i<2;i++){
if(i==0) aa[i][i+1]=*p+1;
else  ++p;
printf("%d",*p);
}
}
```
的输出是_____。
A：23
B：26
C：33
D：36

15、若定义了以下函数：
```
void f(……)
{……
*p=(double*)malloc(10*sizeof(double));
……
}
```
p是该函数的形参，要求通过p把动态分配存储单元的地址传回主调函数，则形参p的正确定义应当是_____。
A：double *p
B：float **p
C：double **p
D：float *p

16、若有说明：int n=2,*p=&n,*q=p;，则以下非法的赋值语句是_____。
A：p=q;

B：*p=*q;
C：n=*q;
D：p=n;

17、有以下程序
```
main()
{ int i,n=0;
for( i=2;i<5;i++)
{do
{ if(i%3) continue;
n++;
}while(!i);
n++;
}
printf("n=%d\n",n);
}
```
程序执行后输出结果是_____。

A：n=5
B：n=2
C：n=3
D：n=4

18、当执行下面程序且输入：ABC时，输出的结果是_____。
```
#include <stdio.h>
#include <string.h>
main( )
{   char ss[10]="12345";
strcat (ss,"6789");
gets(ss);printf("%s\n",ss);
}
```

A：ABC
B：ABC9
C：123456ABC
D：ABC456789

19、假定int类型变量占用两个字节，若有定义:int x[10]={0,2,4};，则数组x在内存中所占字节数是_____。
A：3
B：6
C：10
D：20

20、以下能正确定义数组并正确赋初值的语句是_____。

A：int N=5,b[N][N];

B：int a[1][2]={{1},{3}};

C：int c[2][]={{1,2},{3,4}};

D：int d[3][2]={{1,2},{3,4}};

C 语言机试测试卷（七）解析

1、答案：D

解析：本题考查的是预处理概念。

通常，预处理命令位于源文件的开头，但不一定必须位于开头，也可以写在函数与函数之间；由于预处理命令的末尾不加分号，所以，不能在一行上写多条预处理命令，否则，系统处理时就会把它当作一条命令；宏名一般习惯用大写字母表示，以便与变量名相区别，但这并非规定，也可用小写字母。

故本题答案为D。

2、答案：B

解析：本题考查的是预处理命令。

选项B正确，原因是这部分语句与C程序其他部分的语句是有区别的，所以在每个预处理语句之前都有一个"#"符号以示区别。C提供三种预处理语句：（1）宏替换，（2）文件包含，（3）条件编译。如果一个文件中要引用另外一个文件时才需要在开头包含预处理命令行：#include<stdio.h>。

故本题答案为B。

3、答案：C

解析：本题的考查点是字符串的长度。

这个语句的功能是输出"ATS\n012\1\\"这个串的长度，在串中"\\"代表一个"\"，为了和printf()函数中的转义字符区分开来，在语法上使用了两个反斜杠代替了一个反斜杠，所以它仅仅为一个字符，而"\1"代表数字1，也占一个字符，"\n"是回车换行符，也占一个字符，加上A,T,S,0,1,2，一共是九个字符。

故本题答案为C。

4、答案：B

解析：本题的考查点是C语言的标识符。

C语言规定，标识符只能由字母、数字和下划线三种符号组成，而且第一个字符必须是字母或下划线。

①中的void是C语言的关键字重名，不合法；

③中的case和C语言的关键字重名，不合法；

④中的2a是数字打头而且sizeof和C语言的关键字重名，不合法。

故本题答案为B。

5、答案：A

解析：本题考查的是常量表示法。

C语言的常量分为整型常量、实型常量和字符型常量。本题中只包含整型常量和实型常量。选项B从形式上来看属于整型常量中的八进制整数，以0开头，但只能用0-7表示八进制数，所以选项B不合法；选项C和D从形式上看属于实型常量，用指数形式表示的实型常量需要注意一点：e（或E）之前必需有数字，且e后面的指数必须为整数，所以选项C和D中e后面为小数和没有数字的形式都不合法；选项A属于实型常量中的十进制小数形式的表示法，是合法的。故本题答案为A。

6、答案：A
解析：本题考查的是条件表达式。
条件表达式的一般形式为：表达式1?表达式2:表达式3
条件运算符的执行顺序为：先求解表达式1，若非0则求解表达式2，此时表达式2的值就是整个条件表达式的值，若表达式1的值为0，则求解表达式3，此时表达式3的值就是整个条件表达式的值。
本题先求的是a<b的值，根据a=1,b=4，故a<b的值为真，所以整个表达式"a<b?a:c<d?c:d"的值为a的值1。即c<d?c:d就不再执行了！
故本题的答案为A。

7、答案：B
解析：本题考查的是数据的输入输出。
对于unsigned型数据可以指定数据的输入域宽w，系统将自动按w值截取所需长度的数据，此题中指定输入域宽为2，所以把前两个数送给a，即a=87；后面的数值应送给变量b，由于6后面是空格分隔符，所以系统认为该数据到此结束，即将6赋给了b。故本题答案为B。

8、答案：A
解析：本题的考查点是逗号表达式。
在(x,y)中的","是一个特殊的运算符，叫做逗号运算符，它的一般形式为：表达式1,表达式2，求解过程为：先求解表达式1，再求解表达式2，整个表达式的值是表达式2的值，(x,y)的值为200，所以输出结果为200。故本题答案为A。

9、解析：本题的考查点是几种运算符的使用。
选项A,'a'&&'b'是字符a与b的相与，故不为0；
选项B,a<=b，由题中变量赋值可知，结果为1。
选项C,a||+c&&b-c，此表达式先做算术运算b-c，结果为-1，而+c属于单目运算符，由于c初值为5，经过单目运算符运算后，还是5，下面再进行逻辑与的运算，即-1&&5结果为1(因为C语言中除0代表假外,其它任一个数都代表真),最后a||1，结果为1。
选项D,!((a<b)&&!c||1),此表达式先运算最外面括号内的表达式(a<b)&&!c||1，然后再进行非运算，由于(a<b)&&!c||1中先算小括号内的a<b结果为1，再按逻辑运算符的运算顺序：!,&&,||，进行运算后得(a<b)&&!c||1的值为1，所以最后再进行非运算知D选项的运算结果为0。
本题答案为D。

10、答案：A

解析：本题考查C语言数据类型。
C语言中长整型数为在数值后加上一个L或l字符。本题答案为A。

11、答案：C
解析：y=012表示将八进制数12赋给变量y。d格式符，用来输出十进制整数。%md，m为指定的输出字段的宽度。如果数据的位数小于m，则左端补以空格，若大于m，则按实际位数输出。本题是将八进制数12也输出为十进制数，八进制数12转换为十进制数为10，所以输出结果为102,10。
故本题答案为C。

12、答案：C
解析：本题的考查点是指针数组的应用。
根据C语言的语法规定可知，int(*ptr)[M]中的标识符ptr是一个指向具有M个整型元素的一维数组的指针。
故本题答案为C。

13、答案：D
解析：本题的考查点是通过指针引用数组元素。
观察程序可知，a实际上就是数组a的首地址，所以"*(a+i)"表示的就是数组a中的第i个元素的值，进而我们可以知道 *(*(a+i))必然不是对a数组元素的正确引用。
故本题答案为D。

14、答案：A
解析：本题的考查点是通过指针引用数组元素。
观察题目，可以发现，*p=&aa[0][0]语句实际是将数组aa的首地址赋给了指针变量p，将i的值带入for循环中，i=0时，aa[0][1]=3，*p=2；i=1时，执行++p语句，*p=3。
故本题答案为A。

15、答案：C
解析：通过程序中给出的语句，我们可以判断，*p是一个指向double型指针的指针。因此，要通过p把动态分配存储单元的地址传回主调函数，应该使用double **p。本题答案为C。

16、答案：D
解析：本题的考查点是指针变量的引用。
指针变量不同于整型变量和其它类型的变量，它是用来存放地址（指针）的，不能将一个整型量（或任何其它非地址类型的数据）赋给一个指针变量，这样的赋值是不合法的。本题中的答案D就属于这种情况，所以p=n;是非法的赋值语句。
故本题答案为D。

17、答案：D
解析：当i=2时，i%3为真，继续执行n++;此时n=1，!i为假，结束while循环，执

行n++，此时n为2。
当i=3时，i%3=0，跳出循环。
当i=4时，i%3=1为真，继续执行n++，此时n为3，!i为假，结束while循环，继续往下执行n++，此时n为4。
故本题答案为D。

18、答案：A
解析：本题的考查点是字符串连接函数strcat()。
strcat()是字符串连接函数，在字符串连接操作以后，有一个重新从键盘取值的语句gets()，所以，当执行下面程序且输入ABC时，程序打印出来的只会是"ABC"。
故本题答案为A。

19、答案：D
解析：本题考查C语言的数组的定义。
C语言中可以为数组赋初值，或部分赋初值。当数组定义后，系统就为其分配内存空间，不论其中有没有内容。本题虽然只给数组x赋了3个初值，但系统仍为数组x分配了10个内存空间。所以数组x在内存中所占字节数为2*10=20。
本题答案为D。

20、答案：D
解析：本题的考查点是二维数组的初始化。
可以用下面方法对二维数组初始化：
1.分行给二维数组赋初值。如
static int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};这种赋初值方法比较直观，把第一个花括弧内的数据赋给第一行的元素，第二个花括弧内的数据赋给第二行的元素，……，即按行赋初值。
2.可以将所有数据写在一个花括弧内，按数组排列的顺序对各元素赋初值。
3.可以对部分元素赋初值。
static int a[3][4]={{1},{5},{9}};
它的作用是只对各行第1列的元素赋初值，其余元素自动为0。
也可以对各行中的某一元素赋初值：static int
a[3][4]={{1},{0,6},{0,0,11}};
也可以只对某几行元素赋初值：static int a[3][4]={{1},{5,6}};
4.如果对全部元素都赋初值，则定义数组时对第一维的长度可以不指定，但第二维的长度不能省。
选项A定义了数组但是没有赋值；
选项B定义的是一行两列，赋值却是两行一列；
选项C在二维数组定义中，行可以不指定，但是列是要指定的。
故本题答案为D。

C 语言机试测试卷（八）

1、Point out the error in the program?
#include<stdio.h>

```
int main(){
        FILE *fp;
        fp=fopen("trial", "r");
        fseek(fp, "20", SEEK_SET);
        fclose(fp);
        return 0;
}
```

A. Error: unrecognised Keyword SEEK_SET
B. Error: fseek() long offset value
C. No error
D. None of above

2、What will be the output of the program (myprog.c) given below if it is executed from the command line?

```
cmd> myprog friday tuesday sunday
/* myprog.c */
#include<stdio.h>
int main(int argc, char *argv[]){
        printf("%c", *++argv[1]);
        return 0;
}
```

A. r
B. f
C. m
D. y

3、What will be the output of the program?

```
#include<stdio.h>
int main(){
        const int x=5;
        const int *ptrx;
        ptrx = &x;
        *ptrx = 10;
        printf("%d\n", x);
        return 0;
}
```

A. 5
B. 10
C. Error
D. Garbage value

4、Point out the error in the program (in Turbo-C).

```
#include<stdio.h>
#define MAX 128
int main(){
    const int max=128;
    char array[max];
    char string[MAX];
    array[0] = string[0] = 'A';
    printf("%c %c\n", array[0], string[0]);
    return 0;
}
```

A. Error: unknown max in declaration/Constant expression required
B. Error: invalid array string
C. None of above
D. No error. It prints A A

5、Input/output function prototypes and macros are defined in which header file?

A. conio.h
B. stdlib.h
C. stdio.h
D. dos.h

6、What will be the output of the program?

```
#include<stdio.h>
int main(){
    int i;
    char c;
    for(i=1; i<=5; i++)
    {
        scanf("%c", &c); /* given input is 'a' */
        printf("%c", c);
        ungetc(c, stdin);
    }
    return 0;
}
```

A. aaaa
B. aaaaa
C. Garbage value.
D. Error in ungetc statement.

7、In which order do the following gets evaluated

1. Relational

2. Arithmetic
3. Logical
4. Assignment
   A. 2134
   B. 1234
   C. 4321
   D. 3214

8、What will be the output of the program?

```
#include<stdio.h>
#define SQUARE(x) x*x
int main(){
      float s=10, u=30, t=2, a;
      a = 2*(s-u*t)/SQUARE(t);
      printf("Result = %f", a);
      return 0;
}
```

   A. Result = -100.000000
   B. Result = -25.000000
   C. Result = 0.000000
   D. Result = 100.000000

9、What will be the output of the program ?

```
#include<stdio.h>
#include<string.h>
int main(){
      char str1[20] = "Hello", str2[20] = " World";
      printf("%s\n", strcpy(str2, strcat(str1, str2)));
      return 0;
}
```

   A. Hello
   B. World
   C. Hello World
   D. WorldHello

10、What will be the output of the program?

```
#define P printf("%d\n", -1^~0);
#define M(P) int main()\
                  {\
                      P\
                      return 0;\
                  }
```

M(P)

A. 1
B. 0
C. -1
D. 2

11、Which of the following range is a valid long double (Turbo C in 16 bit DOS OS) ?

A. 3.4E-4932 to 1.1E+4932
B. 3.4E-4932 to 3.4E+4932
C. 1.1E-4932 to 1.1E+4932
D. 1.7E-4932 to 1.7E+4932

12、What will be the output of the program?

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int i=0;
    i++;
    if(i<=5)
    {
        printf("IndiaBIX");
        exit(1);
        main();
    }
    return 0;
}
```

A. Prints "IndiaBIX" 5 times
B. Function main() doesn't calls itself
C. Infinite loop
D. Prints "IndiaBIx"

13、What will be the output of the program if the array begins 1200 in memory?

```
#include<stdio.h>
int main(){
    int arr[]={2, 3, 4, 1, 6};
    printf("%u, %u, %u\n", arr, &arr[0], &arr);
    return 0;
}
```

A. 1200, 1202, 1204
B. 1200, 1200, 1200

C. 1200, 1204, 1208
D. 1200, 1202, 1200

14、What will be the output of the program ?

```
#include<stdio.h>
int main(){
        FILE *ptr;
        char i;
        ptr = fopen("myfile.c", "r");
        while((i=fgetc(ptr))!=NULL)
                printf("%c", i);
        return 0;
}
```

A. Print the contents of file "myfile.c"
B. Print the contents of file "myfile.c" upto NULL character
C. Infinite loop
D. Error in program

15、Point out the error in the following program.

```
#include<stdio.h>
#include<string.h>
int main(){
        char str1[] = "Learn through IndiaBIX\0.com",    str2[120];
        char *p;
        p = (char*) memccpy(str2, str1, 'i', strlen(str1));
        *p = '\0';
        printf("%s", str2);
        return 0;
}
```

A. Error: in memccpy statement
B. Error: invalid pointer conversion
C. Error: invalid variable declaration
D. No error and prints "Learn through Indi"

16、What will be the output of the program?

```
#include<stdio.h>
#include<math.h>
int main(){
        float n=1.54;
        printf("%f, %f\n", ceil(n), floor(n));
        return 0;
}
```

A. 2.000000, 1.000000
B. 1.500000, 1.500000
C. 1.550000, 2.000000
D. 1.000000, 2.000000

17、Which of the following statements correct about the below program?

```
#include<stdio.h>
int main(){
    union a
    {
        int i;
        char ch[2];
    };
    union a u1 = {512};
    union a u2 = {0, 2};
    return 0;
}
```

1: u2 CANNOT be initialized as shown.
2: u1 can be initialized as shown.
3: To initialize char ch[] of u2 '.' operator should be used.
4: The code causes an error 'Declaration syntax error'

A. 1, 2
B. 2, 3
C. 1, 2, 3
D. 1, 3, 4

18、What will be the output of the program?

```
#include<stdio.h>
int main(){
    char huge *near *far *ptr1;
    char near *far *huge *ptr2;
    char far *huge *near *ptr3;
    printf("%d, %d, %d\n", sizeof(**ptr1), sizeof(ptr2), sizeof(*ptr3));
    return 0;
}
```

A. 4, 4, 4
B. 2, 2, 2
C. 2, 8, 4
D. 2, 4, 8

19、Point out the error in the following program.

```
#include<stdio.h>
```

```
void display(int (*ff)());
int main(){
        int show();
        int (*f)();
        f = show;
        display(f);
        return 0;
}
void display(int (*ff)()){
        (*ff)();
}
int show(){
        printf("IndiaBIX");
}
```

A. Error: invalid parameter in function display()
B. Error: invalid function call f=show;
C. No error and prints "IndiaBIX"
D. No error and prints nothing.

20、Point out the correct statements are correct about the program below?

```
#include<stdio.h>
int main(){
        char ch;
        while(x=0;x<=255;x++)
                printf("ASCII value of %d character %c\n", x, x);
        return 0;
}
```

A. The code generates an infinite loop
B. The code prints all ASCII values and its characters
C. Error: x undeclared identifier
D. Error: while statement missing

C 语言机试测试卷（八）解析

1、Correct Answer: Option B
Explanation:
Instead of "20" use 20L since fseek() need a long offset value.

2、Correct Answer: Option A

3、Correct Answer: Option C
Explanation:

Step 1: const int x=5; The constant variable x is declared as an integer data type and initialized with value '5'.
Step 2: const int *ptrx; The constant variable ptrx is declared as an integer pointer.
Step 3: ptrx = &x; The address of the constant variable x is assigned to integer pointer variable ptrx.
Step 4: *ptrx = 10; Here we are indirectly trying to change the value of the constant vaiable x. This will result in an error.
To change the value of const variable x we have to use *(int *)&x = 10;

4、Correct Answer: Option A
Explanation:
Step 1: A macro named MAX is defined with value 128
Step 2: const int max=128; The constant variable max is declared as an integer data type and it is initialized with value 128.
Step 3: char array[max]; This statement reports an error "constant expression required". Because, we cannot use variable to define the size of array.
To avoid this error, we have to declare the size of an array as static. Eg. char array[10]; or use macro char array[MAX];
Note: The above program will print A A as output in Unix platform.

5、Correct Answer: Option C
Explanation:
stdio.h, which stands for "standard input/output header", is the header in the C standard library that contains macro definitions, constants, and declarations of functions and types used for various standard input and output operations.

6、Correct Answer: Option B
Explanation:
for(i=1; i<=5; i++) Here the for loop runs 5 times.
Loop 1:
scanf("%c", &c); Here we give 'a' as input.
printf("%c", c); prints the character 'a' which is given in the previous "scanf()" statement.
ungetc(c, stdin); "ungetc()" function pushes character 'a' back into input stream.
Loop 2:
Here the scanf("%c", &c); get the input from "stdin" because of "ungetc" function.
printf("%c", c); Now variable c = 'a'. So it prints the character 'a'.

ungetc(c, stdin); "ungetc()" function pushes character 'a' back into input stream.
This above process will be repeated in Loop 3, Loop 4, Loop 5.

7、Correct Answer: Option A
Explanation:
2. Arithmetic operators: *, /, %, +, -
1. Relational operators: >, <, >=, <=, ==, !=
3. Logical operators : !, &&, ||
4. Assignment operators: =

8、Correct Answer: Option A
Explanation:
The macro function SQUARE(x) x*x calculate the square of the given number 'x'. (Eg: 102)
Step 1: float s=10, u=30, t=2, a; Here the variable s, u, t, a are declared as an floating point type and the variable s, u, t are initialized to 10, 30, 2.
Step 2: a = 2*(s-u*t)/SQUARE(t); becomes,
=> a = 2 * (10 - 30 * 2) / t * t; Here SQUARE(t) is replaced by macro to t*t .
=> a = 2 * (10 - 30 * 2) / 2 * 2;
=> a = 2 * (10 - 60) / 2 * 2;
=> a = 2 * (-50) / 2 * 2 ;
=> a = 2 * (-25) * 2 ;
=> a = (-50) * 2 ;
=> a = -100;
Step 3: printf("Result=%f", a); It prints the value of variable 'a'.
Hence the output of the program is -100

9、Correct Answer: Option C
Explanation:
Step 1: char str1[20] = "Hello", str2[20] = " World"; The variable str1 and str2 is declared as an array of characters and initialized with value "Hello" and " World" respectively.
Step 2: printf("%s\n", strcpy(str2, strcat(str1, str2)));
=> strcat(str1, str2)) it append the string str2 to str1. The result will be stored in str1. Therefore str1 contains "Hello World".
=> strcpy(str2, "Hello World") it copies the "Hello World" to the variable str2.
Hence it prints "Hello World".

10、Correct Answer: Option B

11、Correct Answer: Option A
Explanation:
The range of long double is 3.4E-4932 to 1.1E+4932

12、Correct Answer: Option D
Explanation:
Step 1: int i=0; The variable i is declared as in integer type and initialized to '0'(zero).
Step 2: i++; Here variable i is increemented by 1. Hence i becomes '1'(one).
Step 3: if(i<=5) becomes if(1 <=5). Hence the if condition is satisfied and it enter into if block statements.
Step 4: printf("IndiaBIX"); It prints "IndiaBIX".
Step 5: exit(1); This exit statement terminates the program execution.
Hence the output is "IndiaBIx".

13、Correct Answer: Option B
Explanation:
Step 1: int arr[]={2, 3, 4, 1, 6}; The variable arr is declared as an integer array and initialized.
Step 2: printf("%u, %u, %u\n", arr, &arr[0], &arr); Here,
The base address of the array is 1200.
=> arr, &arr is pointing to the base address of the array arr.
=> &arr[0] is pointing to the address of the first element array arr. (ie. base address)
Hence the output of the program is 1200, 1200, 1200

14、Correct Answer: Option C
Explanation:
The program will generate infinite loop. When an EOF is encountered fgetc() returns EOF. Instead of checking the condition for EOF we have checked it for NULL. so the program will generate infinite loop.

15、Correct Answer: Option D
Explanation:
Declaration:
void *memccpy(void *dest, const void *src, int c, size_t n); : Copies a block of n bytes from src to dest
With memccpy(), the copying stops as soon as either of the following occurs:
=> the character 'i' is first copied into str2

=> n bytes have been copied into str2

16、Correct Answer: Option A
Explanation:
ceil(x) round up the given value. It finds the smallest integer not < x.
floor(x) round down the given value. It finds the smallest integer not > x.
printf("%f, %f\n", ceil(n), floor(n)); In this line ceil(1.54) round up the 1.54 to 2 and floor(1.54) round down the 1.54 to 1.
In the printf("%f, %f\n", ceil(n), floor(n)); statement, the format specifier "%f %f" tells output to be float value. Hence it prints 2.000000 and 1.000000.

17、Correct Answer: Option C

18、Correct Answer: Option A

19、Correct Answer: Option C

20、Correct Answer: Option D
Explanation:
There are 2 errors in this program.
1. "Undefined symbol x" error. Here x is not defined in the program.
2. Here while() statement syntax error.

C 语言机试测试卷（九）

1、What will be the output of the program ?

```
#include<stdio.h>
int main(){
    int i;
    char a[] = "\0";
    if(printf("%s", a))
        printf("The string is empty\n");
    else
        printf("The string is not empty\n");
    return 0;
}
```

A. The string is empty
B. The string is not empty
C. No output
D. 0

2、What will be the output of the following program in 16 bit platform assuming that 1022 is memory address of the string "Hello1" (in Turbo C under DOS) ?

```
#include<stdio.h>
int main(){
        printf("%u %s\n", &"Hello1", &"Hello2");
        return 0;
}
```

A. 1022 Hello2
B. Hello1 1022
C. Hello1 Hello2
D. 1022 1022
E. Error

3、What will be the output of the program?

```
#include<stdio.h>
int fun(int **ptr);
int main(){
        int i=10;
        const int *ptr = &i;
        fun(&ptr);
        return 0;
}
int fun(int **ptr){
        int j = 223;
        int *temp = &j;
        printf("Before changing ptr = %5x\n", *ptr);
        const *ptr = temp;
        printf("After changing ptr = %5x\n", *ptr);
        return 0;
}
```

A. Address of i Address of j
B. 10 223
C. Error: cannot convert parameter 1 from 'const int **' to 'int **'
D. Garbage value

4、What will be the output of the program?

```
#include<stdio.h>
#include<stdlib.h>
int main(){
        char *i = "55.555";
        int result1 = 10;
```

```
    float result2 = 11.111;
    result1 = result1+atoi(i);
    result2 = result2+atof(i);
    printf("%d, %f", result1, result2);
    return 0;
}
```

A. 55, 55.555
B. 66, 66.666600
C. 65, 66.666000
D. 55, 55

5、Which of the following cannot be checked in a switch-case statement?

A. Character
B. Integer
C. Float
D. enum

6、Point out the error, if any in the program.

```
#include<stdio.h>
int main(){
    int a = 10, b;
    a >=5 ? b=100: b=200;
    printf("%d\n", b);
    return 0;
}
```

A. 100
B. 200
C. Error: L value required for b
D. Garbage value

7、Which of the following statements are correct about the program?

```
#include<stdio.h>
int main(){
    int x = 30, y = 40;
    if(x == y)
        printf("x is equal to y\n");

    else if(x > y)
        printf("x is greater than y\n");

    else if(x < y)
        printf("x is less than y\n")
```

```
    return 0;
}
```

A. Error: Statement missing
B. Error: Expression syntax
C. Error: Lvalue required
D. Error: Rvalue required

8、What will be the output of the program?

```
#include<stdio.h>
int main(){
    int i=4, j=-1, k=0, w, x, y, z;
    w = i || j || k;
    x = i && j && k;
    y = i || j &&k;
    z = i && j || k;
    printf("%d, %d, %d, %d\n", w, x, y, z);
    return 0;
}
```

A. 1, 1, 1, 1
B. 1, 1, 0, 1
C. 1, 0, 0, 1
D. 1, 0, 1, 1

9、Is this a correct way for NULL pointer assignment?

```
int i=0;
char *q=(char*)i;
```

A. Yes
B. No

10、On executing the below program what will be the contents of 'target.txt' file if the source file contains a line "To err is human"?

```
#include<stdio.h>
int main(){
    int i, fss;
    char ch, source[20] = "source.txt", target[20]="target.txt", t;
    FILE *fs, *ft;
    fs = fopen(source, "r");
    ft = fopen(target, "w");
    while(1)
    {
        ch=getc(fs);
        if(ch==EOF)
```

```
                break;
            else
            {
                fseek(fs, 4L, SEEK_CUR);
                fputc(ch, ft);
            }
        }
        return 0;
}
```

A. r n
B. Trh
C. err
D. None of above

11、What will be the output of the program (sample.c) given below if it is executed from the command line (Turbo C in DOS)?

```
cmd> sample 1 2 3
/* sample.c */
#include<stdio.h>
int main(int argc, char *argv[]){
        int j;
        j = argv[1] + argv[2] + argv[3];
        printf("%d", j);
        return 0;
}
```

A. 6
B. sample 6
C. Error
D. Garbage value

12、How many bytes are occupied by near, far and huge pointers (DOS)?

A. near=2 far=4 huge=4
B. near=4 far=8 huge=8
C. near=2 far=4 huge=8
D. near=4 far=4 huge=8

13、Which of the statements is correct about the program?

```
#include<stdio.h>
int main(){
        int arr[3][3] = {1, 2, 3, 4};
        printf("%d\n", *(*(*(arr))));
        return 0;
```

}

A. Output: Garbage value
B. Output: 1
C. Output: 3
D. Error: Invalid indirection

14、What will be the output of the program in Turbo C (under DOS)?

```
#include<stdio.h>
int main(){
    struct emp
    {
        char *n;
        int age;
    };
    struct emp e1 = {"Dravid", 23};
    struct emp e2 = e1;
    strupr(e2.n);
    printf("%s\n", e1.n);
    return 0;
}
```

A. Error: Invalid structure assignment
B. DRAVID
C. Dravid
D. No output

15、Point out the error in the program?

```
#include<stdio.h>
int main(){
    struct a
    {
        float category:5;
        char scheme:4;
    };
    printf("size=%d", sizeof(struct a));
    return 0;
}
```

A. Error: invalid structure member in printf
B. Error in this float category:5; statement
C. No error
D. None of above

16、What is the purpose of fflush() function.

A. flushes all streams and specified streams.
B. flushes only specified stream.
C. flushes input/output buffer.
D. flushes file buffer.

17、What will be the output of the program, if a short int is 2 bytes wide?

```
#include<stdio.h>
int main(){
    short int i = 0;
    for(i<=5 && i>=-1; ++i; i>0)
        printf("%u,", i);
    return 0;
}
```

A. 1 ... 65535
B. Expression syntax error
C. No output
D. 0, 1, 2, 3, 4, 5

18、What will be the output of the program?

```
#include<stdio.h>
int main(){
    float a = 0.7;
    if(0.7 > a)
        printf("Hi\n");
    else
        printf("Hello\n");
    return 0;
}
```

A. Hi
B. Hello
C. Hi Hello
D. None of above

19、Which of the following function sets first n characters of a string to a given character?

A. strinit()
B. strnset()
C. strset()
D. strcset()

20、If the size of pointer is 4 bytes then What will be the output of the program ?

```
#include<stdio.h>
int main(){
        char *str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"};
        printf("%d, %d", sizeof(str), strlen(str[0]));
        return 0;
}
```

A. 22, 4
B. 25, 5
C. 24, 5
D. 20, 2

C 语言机试测试卷（九）解析

1、Correct Answer: Option B
Explanation:
The function printf() returns the number of charecters printed on the console.
Step 1: char a[] = "\0"; The variable a is declared as an array of characters and it initialized with "\0". It denotes that the string is empty.
Step 2: if(printf("%s", a)) The printf() statement does not print anything, so it returns '0'(zero). Hence the if condition is failed.
In the else part it prints "The string is not empty".

2、Correct Answer: Option A
Explanation:
In printf("%u %s\n", &"Hello", &"Hello");.
The %u format specifier tells the compiler to print the memory address of the "Hello1".
The %s format specifier tells the compiler to print the string "Hello2".
Hence the output of the program is "1022 Hello2".

3、Correct Answer: Option C

4、Correct Answer: Option C
Explanation:
Function atoi() converts the string to integer.
Function atof() converts the string to float.
result1 = result1+atoi(i);
Here result1 = 10 + atoi(55.555);
result1 = 10 + 55;
result1 = 65;
result2 = result2+atof(i);

Here result2 = 11.111 + atof(55.555);
result2 = 11.111 + 55.555000;
result2 = 66.666000;
So the output is "65, 66.666000" .

5、Correct Answer: Option C
Explanation:
The switch/case statement in the c language is defined by the language specification to use an int value, so you can not use a float value.

```
switch( expression ){
    case constant-expression1:    statements 1;
    case constant-expression2:    statements 2;
    case constant-expression3:    statements3 ;
    ...
    ...
    default : statements 4;
}
```

The value of the 'expression' in a switch-case statement must be an integer, char, short, long. Float and double are not allowed.

6、Correct Answer: Option C
Explanation:
Variable b is not assigned.
It should be like:
b = a >= 5 ? 100 : 200;

7、Correct Answer: Option A
Explanation:
This program will result in error "Statement missing ;"
printf("x is less than y\n") here ; should be added to the end of this statement.

8、Correct Answer: Option D
Explanation:
Step 1: int i=4, j=-1, k=0, w, x, y, z; here variable i, j, k, w, x, y, z are declared as an integer type and the variable i, j, k are initialized to 4, -1, 0 respectively.
Step 2: w = i || j || k; becomes w = 4 || -1 || 0;. Hence it returns TRUE. So, w=1
Step 3: x = i && j && k; becomes x = 4 && -1 && 0; Hence it returns FALSE. So, x=0
Step 4: y = i || j &&k; becomes y = 4 || -1 && 0; Hence it returns TRUE.

So, y=1
Step 5: z = i && j || k; becomes z = 4 && -1 || 0; Hence it returns TRUE.
So, z=1.
Step 6: printf("%d, %d, %d, %d\n", w, x, y, z); Hence the output is "1, 0, 1, 1".

9、Correct Answer: Option B
Explanation:
The correct way is char *q=0 (or) char *q=(char*)0

10、Correct Answer: Option B
Explanation:
The file source.txt is opened in read mode and target.txt is opened in write mode. The file source.txt contains "To err is human".
Inside the while loop,
ch=getc(fs); The first character('T') of the source.txt is stored in variable ch and it's checked for EOF.
if(ch==EOF) If EOF(End of file) is true, the loop breaks and program execution stops.
If not EOF encountered, fseek(fs, 4L, SEEK_CUR); the file pointer advances 4 character from the current position. Hence the file pointer is in 5th character of file source.txt.
fputc(ch, ft); It writes the character 'T' stored in variable ch to target.txt.
The while loop runs three times and it write the character 1st and 5th and 11th characters ("Trh") in the target.txt file.

11、Correct Answer: Option C
Explanation:
Here argv[1], argv[2] and argv[3] are string type. We have to convert the string to integer type before perform arithmetic operation.
Example: j = atoi(argv[1]) + atoi(argv[2]) + atoi(argv[3]);

12、Correct Answer: Option A
Explanation:
near=2, far=4 and huge=4 pointers exist only under DOS. Under windows and Linux every pointers is 4 bytes long.

13、Correct Answer: Option D

14、Correct Answer: Option B

15、Correct Answer: Option B

Explanation:
Bit field type must be signed int or unsigned int.
The char type: char scheme:4; is also a valid statement.

16、Correct Answer: Option A
Explanation:
"fflush()" flush any buffered output associated with filename, which is either a file opened for writing or a shell command for redirecting output to a pipe or coprocess.
Example:
fflush(FilePointer);
fflush(NULL); flushes all streams.

17、Correct Answer: Option A
Explanation:
for(i<=5 && i>=-1; ++i; i>0) so expression i<=5 && i>=-1 initializes for loop. expression ++i is the loop condition. expression i>0 is the increment expression.
In for(i <= 5 && i >= -1; ++i; i>0) expression i<=5 && i>=-1 evaluates to one.
Loop condition always get evaluated to true. Also at this point it increases i by one.
An increment_expression i>0 has no effect on value of i.so for loop get executed till the limit of integer (ie. 65535)

18、Correct Answer: Option A
Explanation:
if(0.7 > a) here a is a float variable and 0.7 is a double constant. The double constant 0.7 is greater than the float variable a. Hence the if condition is satisfied and it prints 'Hi'
Example:
```
#include<stdio.h>
int main(){
	float a=0.7;
	printf("%.10f %.10f\n",0.7, a);
	return 0;
}
```
Output:
0.7000000000 0.6999999881

19、Correct Answer: Option B
Explanation:

Declaration:
char *strnset(char *s, int ch, size_t n); Sets the first n characters of s to ch

```
#include <stdio.h>
#include <string.h>
int main(void){
    char *string = "abcdefghijklmnopqrstuvwxyz";
    char letter = 'x';
    printf("string before strnset: %s\n", string);
    strnset(string, letter, 13);
    printf("string after    strnset: %s\n", string);
    return 0;
}
```

Output:
string before strnset: abcdefghijklmnopqrstuvwxyz
string after strnset: xxxxxxxxxxxxxnopqrstuvwxyz

20、Correct Answer: Option C
Explanation:
Step 1: char *str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"};
The variable str is declared as an pointer to the array of 6 strings.
Step 2: printf("%d, %d", sizeof(str), strlen(str[0]));
sizeof(str) denotes 6 * 4 bytes = 24 bytes. Hence it prints '24'
strlen(str[0])); becomes strlen(Frogs)). Hence it prints '5';
Hence the output of the program is 24, 5
Hint: If you run the above code in 16 bit platform (Turbo C under DOS) the output will be 12, 5. Because the pointer occupies only 2 bytes. If you run the above code in Linux (32 bit platform), the output will be 24, 5 (because the size of pointer is 4 bytes).

C 语言机试测试卷（十）

1、Point out the error in the following program.

```
#include<stdio.h>
int main(){
    struct emp{
        char name[20];
        float sal;
    };
    struct emp e[10];
    int i;
    for(i=0; i<=9; i++)
        scanf("%s %f", e[i].name, &e[i].sal);
    return 0;
```

}

A. Suspicious pointer conversion
B. Floating point formats not linked (Run time error)
C. Cannot use scanf() for structures
D. Strings cannot be nested inside structures

2、The library function used to find the last occurrence of a character in a string is

A. strnstr()
B. laststr()
C. strrchr()
D. strstr()

3、If char=1, int=4, and float=4 bytes size, What will be the output of the program ?

```
#include<stdio.h>
int main(){
    char ch = 'A';
    printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14f));
    return 0;
}
```

A. 1, 2, 4
B. 1, 4, 4
C. 2, 2, 4
D. 2, 4, 8

4、What will be the output of the program ?

```
#include<stdio.h>
int main(){
    char str[] = "Nagpur";
    str[0]='K';
    printf("%s, ", str);
    str = "Kanpur";
    printf("%s", str+1);
    return 0;
}
```

A. Kagpur, Kanpur
B. Nagpur, Kanpur
C. Kagpur, anpur
D. Error

5、Which header file should you include, if you are going to develop a

function, which can accept variable number of arguments?

A. varagrg.h
B. stdlib.h
C. stdio.h
D. stdarg.h

Correct Answer: Option D

6、Which of the following errors would be reported by the compiler on compiling the program given below?

```
#include<stdio.h>
int main(){
    int a = 5;
    switch(a)
    {
    case 1:
    printf("First");
    case 2:
    printf("Second");
    case 3 + 2:
    printf("Third");
    case 5:
    printf("Final");
    break;
    }
    return 0;
}
```

A. There is no break statement in each case.
B. Expression as in case 3 + 2 is not allowed.
C. Duplicate case case 5:
D. No error will be reported.

7、What will be the output of the program?

```
#include<stdio.h>
int main(){
    int i=2;
    printf("%d, %d\n", ++i, ++i);
    return 0;
}
```

A. 3, 4
B. 4, 3
C. 4, 4
D. Output may vary from compiler to compiler

8、What is the notation for following functions?

1.
```
int f(int a, float b){
        /* Some code */
}
```

2.
```
int f(a, b)
 int a; float b;
 {
        /* Some code */
 }
```

A. 1. KR Notation 2. ANSI Notation
B. 1. Pre ANSI C Notation 2. KR Notation
C. 1. ANSI Notation 2. KR Notation
D. 1. ANSI Notation 2. Pre ANSI Notation

9、How many times the program will print "IndiaBIX" ?

```
#include<stdio.h>
int main(){
    printf("IndiaBIX");
    main();
    return 0;
}
```

A. Infinite times
B. 32767 times
C. 65535 times
D. Till stack overflows

10、What will be the output of the program?

```
#include<stdio.h>
#define SQR(x)(x*x)
int main(){
    int a, b=3;
    a = SQR(b+2);
    printf("%d\n", a);
    return 0;
}
```

A. 25
B. 11
C. Error
D. Garbage value

11、In a file contains the line "I am a boy\r\n" then on reading this line

into the array str using fgets(). What will str contain?

A. "I am a boy\r\n\0"
B. "I am a boy\r\0"
C. "I am a boy\n\0"
D. "I am a boy"

12、If the file 'source.txt' contains a line "Be my friend" which of the following will be the output of below program?

```
#include<stdio.h>
int main(){
    FILE *fs, *ft;
    char c[10];
    fs = fopen("source.txt", "r");
    c[0] = getc(fs);
    fseek(fs, 0, SEEK_END);
    fseek(fs, -3L, SEEK_CUR);
    fgets(c, 5, fs);
    puts(c);
    return 0;
}
```

A. friend
B. frien
C. end
D. Error in fseek();

13、Point out the correct statement which correctly allocates memory dynamically for 2D array following program?

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int *p, i, j;
    /* Add statement here */
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++)
        {
            p[i*4+j] = i;
            printf("%d", p[i*4+j]);
        }
    }
    return 0;
}
```

A. p = (int*) malloc(3, 4);

B. p = (int*) malloc(3*sizeof(int));
C. p = malloc(3*4*sizeof(int));
D. p = (int*) malloc(3*4*sizeof(int));

14、Which statement will you add in the following program to work it correctly?

```
#include<stdio.h>
int main(){
    printf("%f\n", log(36.0));
    return 0;
}
```

A. #include<conio.h>
B. #include<math.h>
C. #include<stdlib.h>
D. #include<dos.h>

15、Which of the following statements correct about k used in the below statement?

```
char ****k;
```

A. k is a pointer to a pointer to a pointer to a char
B. k is a pointer to a pointer to a pointer to a pointer to a char
C. k is a pointer to a char pointer
D. k is a pointer to a pointer to a char

16、What will be the output of the program in 16-bit platform (Turbo C under DOS) ?

```
#include<stdio.h>
int main(){
    printf("%d, %d, %d", sizeof(3.0f), sizeof('3'), sizeof(3.0));
    return 0;
}
```

A. 8, 1, 4
B. 4, 2, 8
C. 4, 2, 4
D. 10, 3, 4

17、Which standard library function will you use to find the last occurance of a character in a string in C?

A. strnchar()
B. strchar()
C. strrchar()
D. strrchr()

18、 How many times "IndiaBIX" is get printed?

```
#include<stdio.h>
int main(){
    int x;
    for(x=-1; x<=10; x++)
    {
        if(x < 5)
            continue;
        else
            break;
        printf("IndiaBIX");
    }
    return 0;
}
```

A. Infinite times
B. 11 times
C. 0 times
D. 10 times

19、What will be the output of the program?

```
#include<stdio.h>
int main(){
    int i=-3, j=2, k=0, m;
    m = ++i && ++j || ++k;
    printf("%d, %d, %d, %d\n", i, j, k, m);
    return 0;
}
```

A. 1, 2, 0, 1
B. -3, 2, 0, 1
C. -2, 3, 0, 1
D. 2, 3, 1, 1

20、What will be the output of the program if the array begins at address 65486?

```
#include<stdio.h>
int main(){
    int arr[] = {12, 14, 15, 23, 45};
    printf("%u, %u\n", arr, &arr);
    return 0;
}
```

A. 65486, 65488

B. 65486, 65486
C. 65486, 65490
D. 65486, 65487

C 语言机试测试卷（十）解析

1、Correct Answer: Option B
Explanation:
Compile and Run the above program in Turbo C:
C:\>myprogram.exe
Sundar
2555.50
scanf : floating point formats not linked
Abnormal program termination
The program terminates abnormally at the time of entering the float value for e[i].sal.

Solution:
Just add the following function in your program. It will force the compiler to include required libraries for handling floating point linkages.

```
static void force_fpf() /* A dummy function */
{
      float x, *y; /* Just declares two variables */
      y = &x;          /* Forces linkage of FP formats */
      x = *y;          /* Suppress warning message about x */
}
```

2、Correct Answer: Option C
Explanation:
Declaration: char *strrchr(const char *s, int c);
It scans a string s in the reverse direction, looking for a specific character c.
Example:

```
#include <string.h>
#include <stdio.h>
int main(void)
{
    char text[] = "I learn through IndiaBIX.com";
    char *ptr, c = 'i';
    ptr = strrchr(text, c);
    if (ptr)
        printf("The position of '%c' is: %d\n", c, ptr-text);
```

```
    else
        printf("The character was not found\n");
    return 0;
}
```
Output:
The position of 'i' is: 19

3、Correct Answer: Option B
Explanation:
Step 1: char ch = 'A'; The variable ch is declared as an character type and initialized with value 'A'.
Step 2:
printf("%d, %d, %d", sizeof(ch), sizeof('A'), sizeof(3.14));
The sizeof function returns the size of the given expression.
sizeof(ch) becomes sizeof(char). The size of char is 1 byte.
sizeof('A') becomes sizeof(65). The size of int is 4 bytes (as mentioned in the question).
sizeof(3.14f). The size of float is 4 bytes.
Hence the output of the program is 1, 4, 4

4、Correct Answer: Option D
Explanation:
The statement str = "Kanpur"; generates the LVALUE required error. We have to use strcpy function to copy a string.
To remove error we have to change this statement str = "Kanpur"; to strcpy(str, "Kanpur");
The program prints the string "anpur"

5、Correct Answer: Option D

6、Correct Answer: Option C
Explanation:
Because, case 3 + 2: and case 5: have the same constant value 5.

7、Correct Answer: Option D
Explanation:
The order of evaluation of arguments passed to a function call is unspecified.
Anyhow, we consider ++i, ++i are Right-to-Left associativity. The output of the program is 4, 3.
In TurboC, the output will be 4, 3.
In GCC, the output will be 4, 4.

8、Correct Answer: Option C
Explanation:
KR Notation means Kernighan and Ritche Notation.

9、Correct Answer: Option D
Explanation:
A call stack or function stack is used for several related purposes, but the main reason for having one is to keep track of the point to which each active subroutine should return control when it finishes executing.

A stack overflow occurs when too much memory is used on the call stack.

Here function main() is called repeatedly and its return address is stored in the stack. After stack memory is full. It shows stack overflow error.

10、Correct Answer: Option B
Explanation:
The macro function SQR(x)(x*x) calculate the square of the given number 'x'. (Eg: 102)

Step 1: int a, b=3; Here the variable a, b are declared as an integer type and the variable b is initialized to 3.

Step 2: a = SQR(b+2); becomes,
=> a = b+2 * b+2; Here SQR(x) is replaced by macro to x*x .
=> a = 3+2 * 3+2;
=> a = 3 + 6 + 2;
=> a = 11;
Step 3: printf("%d\n", a); It prints the value of variable 'a'.
Hence the output of the program is 11

11、Correct Answer: Option C
Explanation:
Declaration: char *fgets(char *s, int n, FILE *stream);
fgets reads characters from stream into the string s. It stops when it reads either n - 1 characters or a newline character, whichever comes first.
Therefore, the string str contain "I am a boy\n\0"

12、Correct Answer: Option C
Explanation:
The file source.txt contains "Be my friend".

fseek(fs, 0, SEEK_END); moves the file pointer to the end of the file.
fseek(fs, -3L, SEEK_CUR); moves the file pointer backward by 3 characters.
fgets(c, 5, fs); read the file from the current position of the file pointer.
Hence, it contains the last 3 characters of "Be my friend".
Therefore, it prints "end".

13、Correct Answer: Option D

14、Correct Answer: Option B
Explanation:
math.h is a header file in the standard library of C programming language designed for basic mathematical operations.
Declaration syntax: double log(double);

15、Correct Answer: Option B

16、Correct Answer: Option B
Explanation:
Step 1:
printf("%d, %d, %d", sizeof(3.0f), sizeof('3'), sizeof(3.0));
The sizeof function returns the size of the given expression.
sizeof(3.0f) is a floating point constant. The size of float is 4 bytes
sizeof('3') It converts '3' in to ASCII value.. The size of int is 2 bytes
sizeof(3.0) is a double constant. The size of double is 8 bytes
Hence the output of the program is 4,2,8
Note: The above program may produce different output in other platform due to the platform dependency of C compiler.
In Turbo C, 4 2 8. But in GCC, the output will be 4 4 8.

17、Correct Answer: Option D
Explanation:
strrchr() returns a pointer to the last occurrence of character in a string.
Example:

```
#include <stdio.h>
#include <string.h>
int main(){
        char str[30] = "12345678910111213";
        printf("The last position of '2' is %d.\n",
                        strrchr(str, '2') - str);
        return 0;
```

}
Output: The last position of '2' is 14.

18、Correct Answer: Option C

19、Correct Answer: Option C
Explanation:
Step 1: int i=-3, j=2, k=0, m; here variable i, j, k, m are declared as an integer type and variable i, j, k are initialized to -3, 2, 0 respectively.
Step 2: m = ++i && ++j || ++k;
becomes m = (-2 && 3) || ++k;
becomes m = TRUE || ++k;.
(++k) is not executed because (-2 && 3) alone return TRUE.
Hence this statement becomes TRUE. So it returns '1'(one). Hence m=1.
Step 3: printf("%d, %d, %d, %d\n", i, j, k, m); In the previous step the value of i,j are increemented by '1'(one).
Hence the output is "-2, 3, 0, 1".

20、Correct Answer: Option B
Explanation:
Step 1: int arr[] = {12, 14, 15, 23, 45}; The variable arr is declared as an integer array and initialized.
Step 2: printf("%u, %u\n", arr, &arr); Here,
The base address of the array is 65486.
=> arr, &arr is pointing to the base address of the array arr.
Hence the output of the program is 65486, 65486

3.1.2 逻辑测试

逻辑测试解读

逻辑测试是和心理测试交叉在一块的，因为时间很少，并且到达规定的时间后，试卷自动就提交了，因此，需要你有很快的逻辑推理能力。试一试下面的题目，题目下面是解析。

逻辑测试卷（一）

1．“假定一切物质都具有在本质上跟感觉相近的特性”的观点，是属于(　)。

A．主观唯心主义观点　　B．辩证唯物主义观点
C．客观唯心主义观点　　D．庸俗唯物主义观点

2．在文学走出去的过程中，用_____的画面表现故事，可以使不同喜好的观众在_____文字、图像、声音的立体式信息空间中，进行文学的超时空阅读和赏析。填入划横线部分最恰当的一项是：

A．千姿百态　包罗　　B．绚丽多彩　充满

C．多姿多彩　融合　　D．风格迥异　调和

3．一个六位数的各位数字都不相同，最左一位数字是 3，且它能被 11 整除，这样的六位数中最小的是(　)。

A．301245　B．301246　C．310254　D．310246

4．黄金标准法则，就是在品牌定位和广告表现上．为品牌设立一个可使之与同类品牌相比更加出色的说辞，从而体现出该品牌的高出一等，胜人一筹；黄金标准可凸显品牌的优越品质、独特利益和全新价值，使品牌的销售主张(USP)更有表现力、冲击力、说服力和促销力。根据上述定义，下列选项中体现了黄金标准法则的是()。

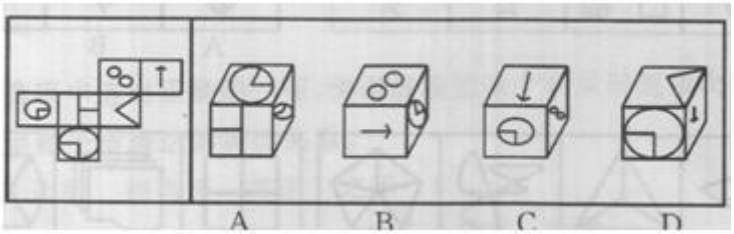
A．品牌的同等配置相机定价要比别的品牌相机低 30%，因此其销量一直很好

B．化妆品公司支使水军发布文章称竞争对手的产品容易导致过敏，使得自己的销量短期上升

C．家具在电视广告中称只要购买自己的家具就有机会获得和该品牌代言明星合影的机会，使得广告期间的家具销售量明显上升

D．衬衫号称自己是衬衫专家，衬衫质量要比其他厂家的好，其市场占有份额一直很高

5.左边给定的是纸盒外表面的展开图，右边哪一项能由它折叠而成?请把它找出来。



6、1，11,21,1211,111221，下一个数字是（）

A．312211　　B.111211

C.211312　　D.122111

7、一个家庭有两个小孩，其中有一个是女孩，问另一个也是女孩的概率是多少？（假定生男生和生女生的概率一样）（）

A.1/2　　B.1/3　　C.1/4　　D.1/5

8、100 个人回答五道试题，有 81 人答对第一题，91 人答对第二题，85 人答对第三题，79 人答对第四题，74 人答对第五题，答对三道题或三道题以上的人算及格， 那么，在这 100 人中，至少有（ ）人及格。

A．50　　B.60　　C.70　　D.80

9、一天,harlan 的店里来了一位顾客,挑了 25 元的货,顾客拿出 100 元,harlan 没零钱找不开,就到隔壁飞白的店里把这 100 元换成零钱,回来给顾客找了 75 元零钱。 过一会,飞白来找 harlan,说刚才的是假钱,harlan 马上给飞白换了张真钱,问 harlan 赔了多少 钱？（ ）

A．25 元　　B.75 元

C.100 元　　D.200 元

10、一个人花 8 块钱买了一只鸡，9 块钱卖掉了，然后他觉得不划算，花 10 块钱又买回来了，11 块钱卖给另外一个人。问他赚了多少钱？

A.0 元　　B.2 元　　C.1 元　　D.4 元

逻辑测试卷（一）解析

1．B 【答案】解析：反映特性是一切物质形态都具有的属性，它是物质形态在同他物的相互作用中复制、再现他物属性并留下“痕迹”的属性和能力。高等动物的长期进化则产生了人类和人的意识。这说明人的意识这种高级反映形式不是凭空突然产生的，是自然界长期发展的产物。因此，此观点是辩证唯物主义观点。

2．C 【答案】解析：先看第二空，“调和”一般和“矛盾”搭配，填在这里不合语境，排除 D。再看第一空，“千姿百态”强调样式多，“绚丽多彩”强调颜色多，“多姿多彩”强调样式和颜色多。用来形容“画面”表现各种风格的故事，用“多姿多彩”更恰当。故本题答案为 C。

3．B 【答案】解析：能被 11 整除的判定规则为：奇数位数字之和与偶数位数字之和的差能被 11 整除，结合选项，只有 B 项符合。

4．D 【解析】A 项，属于利用低价格营销，B 项，属于不正当营销手段，C 项是利用明星来营销，并不能凸显商品本身的优越品质、独特利益和全新价值，而 D 项的说辞则凸显了自己产品的优越品质，故本题选择 D 项。

5. A 【解析】B 项，右侧面的内部图形应为不与边相切的圆；C 项右侧面两个小圆的方向应在另一条对角线上，D 项中正面内切圆的横向半径线应指向右边。故本题选 A。

6.A【解析】A 项，下行是对上一行的解释，如 1,11 中第二个数字 11 就表示 1 个 1；所以新的数应该是 312211，表示：3 个 1 2 个 2 1 个 1

7.B【解析】样本空间为（男男）（女女）（男女）（女男）
A＝ （已知其中一个是女孩）＝ （女女）（男女）（女男）
B＝ （另一个也是女孩）＝ （女女）
于是 P(B/A) = P(AB)/P(A)=(1/4)/(3/4)=1/3

8．C【解析】首先求解原题。每道题的答错人数为（次序不重要）：26，21，19，15，9
第 3 分布层：答错 3 道题的最多人数为：（26+21+19+15+9）/3=30
第 2 分布层：答错 2 道题的最多人数为：（21+19+15+9）/2=32
第 1 分布层：答错 1 道题的最多人数为：（19+15+9）/1=43
Max_3=Min(30, 32, 43)=30。因此答案为：100-30=70。
其实，因为 26 小于 30，所以在求出第一分布层后，就可以判断答案为 70 了。 要让及格的人数最少，就要做到两点：
1. 不及格的人答对的题目尽量多，这样就减少了及格的人需要答对的题目的数量，也就只需要更少的及格的人
2. 每个及格的人答对的题目数尽量多，这样也能减少及格的人数
由 1 得每个人都至少做对两道题目
由 2 得要把剩余的 210 道题目分给其中的 70 人： 210/3 = 70，让这 70 人全部题目都做对，而其它 30 人只做对了两道题
也很容易给出一个具体的实现方案：
让 70 人答对全部五道题，11 人仅答对第一、二道题，10 人仅答对第二、三道题，5 人答对第三、四道题，4 人仅答对第四、五道题
显然稍有变动都会使及格的人数上升。所以最少及格人数就是 70 人！

9.C 【解析】25 元的货和 75 元零钱全都包含在顾客拿出的 100 元假钞里面，所以 harlan 给飞白换成真钱赔了 100 元。

10.B 【解析】总共赚取 2 元。题目很简单，主要是锻炼你的信息筛选能力，买卖两次，每次赚取 1 元，所以总共是 2 元。

逻辑测试卷（二）

1、有一个大蛋糕,用刀平整地切,总共切 7 刀,最多能切成____份,最少能切成____份 ?

2、一楼到十楼的每层电梯门口都放着一颗钻石，钻石大小不一。你乘坐电梯从一楼到十楼，每层楼电梯门都会打开一次，只能拿一次钻石，问怎样才能拿到最大的一颗?

3、有 7 克、2 克砝码各一个，天平一只，如何只用这些物品三次将 140 克的盐分成 50、90 克各一份?

4、小明和小强都是张老师的学生,张老师的生日是 M 月 N 日,2 人都知道张老师的生日是下列 10 组中的一天,张老师把 M 值告诉了小明,把 N 值告诉了小强,张老师问他们知道他的生日是那一天吗?
3 月 4 日 3 月 5 日 3 月 8 日
6 月 4 日 6 月 7 日
9 月 1 日 9 月 5 日
12 月 1 日 12 月 2 日 12 月 8 日
小明说：如果我不知道的话,小强肯定也不知道
小强说：本来我也不知道,但是现在我知道了
小明说：哦,那我也知道了
请根据以上对话推断出张老师的生日是哪一天

5、说从前啊,有一个富人,他有 30 个孩子,其中 15 个是已故的前妻所生,其余 15 个是继室所生,这后一个妇人很想让她自己所生的最年长的儿子继承财产,于是,有一天,他就向他说 :"亲爱的丈夫啊,你就要老了,我们应该定下来谁将是你的继承人,让我们把我们的 30 个孩子排成一个圆圈,从他们中的一个数起,每逢到 10 就让那个孩子站出去,直到最后剩下哪个孩子,哪个孩子就继承你的财产吧!"富人一想,我靠,这个题意相当有内涵了,不错,仿佛很公平,就这么办吧~不过,当剔选过程不断进行下去的时候,这个富人傻眼了,他发现前 14 个被剔除的孩子都是前妻生的,而且下一个要被剔除的还是前妻生的,富人马上大手一挥,停,现在从这个孩子倒回去数,继室,就是这个歹毒的后妈一想,倒数就倒数,我 15 个儿子还斗不过你一个啊~她立即同意了富人的动议,你猜,到底谁做了继承人呢~

6、在一张长方形的桌面上放了 n 个一样大小的圆形硬币。这些硬币中可能有一些不完全在桌面内，也可能有一些彼此重叠；当再多放一个硬币而它的圆心在桌面内时，新放的硬币便必定与原先某些硬币重叠。请证明整个桌面可以用 4n 个硬币完全覆盖。

7、前提：
1 有五栋五种颜色的房子
2 每一位房子的主人国籍都不同
3 这五个人每人只喝一种饮料，只抽一种牌子的香烟，只养一种宠物
4 没有人有相同的宠物，抽相同牌子的香烟，喝相同的饮料
提示：1 英国人住在红房子里
2 瑞典人养了一条狗
3 丹麦人喝茶
4 绿房子在白房子左边
5 绿房子主人喝咖啡
6 抽ＰＡＬＬ　ＭＡＬＬ烟的人养了一只鸟
7 黄房子主人抽ＤＵＮＨＩＬＬ烟
8 住在中间那间房子的人喝牛奶
9 挪威人住第一间房子
10 抽混合烟的人住在养猫人的旁边
11 养马人住在抽ＤＵＮＨＩＬＬ烟的人旁边
12 抽ＢＬＵＥ　ＭＡＳＴＥＲ烟的人喝啤酒
13 德国人抽ＰＲＩＮＣＥ烟
14 挪威人住在蓝房子旁边
15 抽混合烟的人的邻居喝矿泉水
问题是：谁养鱼？？？

8、5 个人来自不同地方，住不同房子，养不同动物，吸不同牌子香烟，喝不同饮料，喜欢不同食物。根据以下线索确定谁是养猫的人。
1. 红房子在蓝房子的右边，白房子的左边（不一定紧邻）
2. 黄房子的主人来自香港，而且他的房子不在最左边。
3. 爱吃比萨的人住在爱喝矿泉水的人的隔壁。
4. 来自北京的人爱喝茅台，住在来自上海的人的隔壁。
5. 吸希尔顿香烟的人住在养马人的右边隔壁。
6. 爱喝啤酒的人也爱吃鸡。
7. 绿房子的人养狗。
8. 爱吃面条的人住在养蛇人的隔壁。
9. 来自天津的人的邻居（紧邻）一个爱吃牛肉，另一个来自成都。
10. 养鱼的人住在最右边的房子里。
11. 吸万宝路香烟的人住在吸希尔顿香烟的人和吸“555”香烟的人的中间（紧邻）

12．红房子的人爱喝茶。

13．爱喝葡萄酒的人住在爱吃豆腐的人的右边隔壁。

14．吸红塔山香烟的人既不住在吸健牌香烟的人的隔壁，也不与来自上海的人相邻。

15．来自上海的人住在左数第二间房子里。

16．爱喝矿泉水的人住在最中间的房子里。

17．爱吃面条的人也爱喝葡萄酒。

18．吸“555”香烟的人比吸希尔顿香烟的人住的靠右

9、烧一根不均匀的绳要用一个小时，如何用它来判断半个小时？烧一根不均匀的绳,从头烧到尾总共需要 1 个小时。现在有若干条材质相同的绳子,问如何用烧绳的方法来计时一个小时十五分钟呢?

10、一个经理有三个女儿, 三个女儿的年龄加起来等于 13,三个女儿的年龄乘起来等于 经理自己的年龄,有一个下属已知道经理的年龄,但仍不能确定经理三个女儿的年龄,这 时经理说只有 一个女儿的头发是黑的,然后这个下属就知道了经理三个女儿的年龄。请问 三个女儿的年龄分别是多少？为什么？

逻辑测试卷（二）解析

1、【解析】最少 8 块。 最多 2^7 块,即 128 块。

2、【解析】先拿下第一楼的钻石，然后在每一楼把手中的钻石与那一楼的钻石相比较，如果那一楼的钻石比手中的钻石大的话那就把手中的钻石换成那一层的钻石。

3、【解析】1、将 140 克用天平平均分成 70 克

2、70 克用天平平均分成 35 克，35+70=105

3、105=50+55，为了得到这个数据，可以用 7 克、2 克的砝码加在天平的两端：50+7=55+2　　55+35=90

4、【解析】答案应该是 9 月 1 日。

1)首先分析这 10 组日期,经观察不难发现,只有 6 月 7 日和 12 月 2 日这两组日期的日数是唯一的。由此可知,如果小强得知的 N 是 7 或者 2,那么他必定知道了老师的生日。

2)再分析“小明说：如果我不知道的话,小强肯定也不知道”,而该 10 组日期的月数分别为 3,6,9,12,而且都相应月的日期都有两组以上,所以小明得知 M 后，是不可能知道老师生日的。

3)进一步分析“小明说：如果我不知道的话,小强肯定也不知道”,结合第 2 步结论,可知小强得知 N 后也绝不可能知道。
4)结合第 3 和第 1 步,可以推断：所有 6 月和 12 月的日期都不是老师的生日,因为如果小明得知的 M 是 6,而若小强的 N==7,则小强就知道了老师的生日。(由第 1 步已经推出),同理,如果小明的 M==12,若小强的 N==2,则小强同样可以知道老师的生日。即：M 不等于 6 和 9。现在只剩下“3 月 4 日， 3 月 5 日， 3 月 8 日， 9 月 1 日， 9 月 5 日”五组日期。而小强知道了,所以 N 不等于 5(有 3 月 5 日和 9 月 5 日),此时, 小强的 N∈(1,4,8)注：此时 N 虽然有三种可能,但对于小强只要知道其中的一种,就得出结论。所以有“小强说：本来我也不知道,但是现在我知道了”, 对于我们则还需要继续推理，至此,剩下的可能是“3 月 4 日， 3 月 8 日， 9 月 1 日”
5)分析“小明说：哦,那我也知道了”,说明 M==9,N==1,(N==5 已经被排除,3 月份的 有两组)

5、【解析】10+11+12+13+14+15+16+17+18+19+20+21+22+23= 198
198/ 30= 6 余 18.　　小孩子站在 18 号位置即可.

6、【解析】要想让新放的硬币不与原先的硬币重叠，两个硬币的圆心距必须大于直径。也就是说，对于桌面上任意一点，到最近的圆心的距离都小于 2，所以，整个桌面可以用 n 个半径为 2 的硬币覆盖。
把桌面和硬币的尺度都缩小一倍，那么，长、宽各是原桌面一半的小桌面，就可以用 n 个半径为 1 的硬币覆盖。那么，把原来的桌子分割成相等的 4 块小桌子，那么每块小桌子都可以用 n 个半径为 1 的硬币覆盖，因此，整个桌面就可以用 4n 个半径为 1 的硬币覆盖。

7、【解析】第一间是黄房子，挪威人住，喝矿泉水，抽 DUNHILL 香烟，养猫；第二间是蓝房子，丹麦人住，喝茶，抽混合烟，养马；第三间是红房子，英国人住，喝牛奶，抽 PALL MALL 烟，养鸟；第四间是绿房子，德国人住，喝咖啡，抽 PRINCE 烟，养猫、马、鸟、狗以外的宠物；第五间是白房子，瑞典人住，喝啤酒，抽 BLUE MASTER 烟，养狗。

8、【解析】第一间是兰房子，住北京人，养马，抽健牌香烟，喝茅台，吃豆腐；第二间是绿房子，住上海人，养狗，抽希尔顿，喝葡萄酒，吃面条第三间是黄房子，住香港人，养蛇，抽万宝路，喝矿泉水，吃牛肉第四间是红房子，住天津人，抽 555，喝茶，吃比萨；第五间是白房子，住成都人，养鱼，抽红塔山，喝啤酒，吃鸡。

9、【解析】一，一根绳子从两头烧，烧完就是半个小时。
二，一根要一头烧，一根从两头烧，两头烧完的时候（30 分），将剩下的一根另一端点着，烧尽就是 45 分钟。再从两头点燃第三根，烧尽就是 1 时 15 分。

10、【解析】显然 3 个女儿的年龄都不为 0,要不爸爸就为 0 岁了,因此女儿的年龄都大于等于 1 岁。这样可以得到下面的情况：
1*1*11=11,　1*2*10=20,　1*3*9=27,　1*4*8=32,　1*5*7=35,
1*6*6= 36,　2*2*9=36,　2*3*8=48,　2*4*7=56,　2*5*6=60,
3*3*7=63,　3*4*6=72,　3*5*5=75,　4 *4*5=80
因为下属已知道经理的年龄,但仍不能确定经理三个女儿的年龄,说明经理是 36 岁 (因为 1*6*6=36,2*2*9=36),所以 3 个女儿的年龄只有 2 种情况,经理又说只有一个 女儿的头发是黑的,说明只有一个女儿是比较大的,其他的都比较小,头发还没有长成黑 色的,所以 3 个女儿的年龄分别为 2, 2, 9！

3.1.3 性格测试

性格测试解读

性格测试没有好坏之分，也没有标准答案，但是要言行一致，前后一致，因为测试的时候题目数量很大，超过 50 道，所以你不可能在短时间内记住每道题，所以保持前后一致最好的方法就是选你内心的想法。下面是示例的题目，如上所说，每个人都是不一样的，也没有答案可以参考，但是可以看看自己能否在 15 分钟内做完。

性格测试卷（一）

1、关于人生观，我的内心其实是：
A、希望能有各种各样的人生体验，所以想法极其多样化
B、在合理的基础上，谨慎确定目标，一旦确定会坚定不移地去做。
C、更加在乎取得一切有可能的成就。
D、毫不喜欢风险，喜欢享受稳定或现状。
2、如果爬山旅游，大多数状况下，在下山回来的路线我最可能：
A、好玩有趣，所以宁愿新路线回巢。
B、安全稳妥，所以宁愿原路线返回。
C、挑战困难，所以宁愿新路线回巢。
D、方便省心，所以宁愿原路线返回。
3、 说话时，我更看重：
A、感觉效果。有时可能会略显得夸张。
B、描述精确。有时可能略过冗长。

C、达成结果。有时可能过于直接让别人不高兴。
D、人际感受。有时可能会不愿讲真话
4、 在大多数时候，我的内心更想要：
A、刺激。经常冒出新点子，想做就做，喜欢与众不同。
B、安全。头脑冷静，不易冲动。
C、挑战。生命中竞赛随处可见，有强烈的“赢”的欲望。
D、稳定。满足自己所拥有的，很少羡慕别人。
5、 我认为自己在情感上的基本特点是：
A、情绪多变，经常波动。
B、外表自我抑制强，但内心感情起伏大，一旦挫伤难以平复。
C、感情不拖泥带水，只是一旦不稳定，容易发怒。
D、天性情绪四平八稳。
6、 我认为自己除了工作外，在控制欲上面，我：
A、没有控制欲，只有感染带动他人的欲望，但自控能力不算强。
B、用规则来保持我对自己的控制和对他人的要求。
C、内心是有控制欲和希望别人服从我的。
D、没兴趣影响别人，也不愿别人来控制我。
7、 当与情人交往时，我最希望对方：
A、经常赞美我，让我享受开心、被关怀且又有一定自由。
B、可随时默契到我内心所想，对我的需求极其敏感。
C、得到对方的认可，我是正确的并且我对其是有价值的。
D、尊重并且相处静谧的。
8、 在人际交往时，我：
A、本质上还是认为与人交往比长时间独处是有乐趣的。
B、非常审慎缓慢地进入，常会被人认为容易有距离感。
C、希望在人际关系中占据主导地位。
D、顺其自然，不温不火，相对被动。
9、 我做事情，经常：
A、缺少长性，不喜欢长期做相同无变化的事情。
B、缺少果断，期待最好的结果但总能先看到事情的不利面。
C、缺少耐性，有时行事过于草率。
D、缺少紧迫，行动迟缓，难下决心。
10、 通常我完成任务的方式是：
A、常赶在最后期限前完成，是临时抱佛脚的高手。
B、自己有严格规定的程序，精确地做，不要麻烦别人。
C、先做，快速做。
D、使用传统的方法按部就班，需要时从他人处得到帮忙。
11、 如果有人深深惹恼我时，我：

A、内心感到受伤，认为没有原谅的可能，可最终很多时候还是会原谅对方。
B、深深感到愤怒，如此之深怎可忘记？我会牢记，同时未来完全避开那个家伙。
C、会火冒三丈，并且内心期望有机会狠狠地回应。
D、避免摊牌，因为还不到那个地步或者自己再去找新朋友。
12、 在人际关系中，我最在意的是：
A、得到他人的赞美和欢迎。
B、得到他人的理解和欣赏。
C、得到他人的感激和尊敬。
D、得到他人的尊重和接纳。
13、在工作上，我表现出来更多的是：
A、充满热忱，有很多想法且很有灵性。
B、心思细腻，完美精确，而且为人可靠。
C、坚强而直截了当，而且有推动力。
D、有耐心，适应性强而且善于协调。
14、 我过往的老师最有可能对我的评价是：
A、情绪起伏大，善于表达和抒发情感。
B、严格保护自己的私密，有时会显得孤独或是不合群。
C、动作敏捷又独立，并且喜欢自己做事情。
D、看起来安稳轻松，反应度偏低，比较温和。
15、 朋友对我的评价最有可能的是：
A、喜欢对朋友述说，也有感染别人的力量。
B、能够提出很多周全的问题，而且需要许多精细的解说。
C、愿意直言想法，有时会直率而犀利地谈论不喜欢的人、事、物。
D、通常与他人一起是倾听者。

性格测试卷（二）

1、 在帮助他人的问题上，我内心的想法是：
A、别人来找我，不太会拒绝，会尽力帮他。
B、值得帮助的人应该帮助。
C、很少承诺要帮，但我若承诺必兑现。
D、虽无英雄打虎胆，常有自告奋勇心。
2、 面对他人对自己的赞美，我内心：
A、没有也无所谓，特别欣喜那也不至于。
B、我不需无关痛痒的赞美，宁可对方欣赏我的能力。
C、思考对方的真实性或立即回避众人的关注。

D、赞美多多益善，总是令人愉悦的。
3、 面对生活，我更像：
A、随和派－外面的世界我无关，我觉得自己这样还不错。
B、行动派－我不进步，别人就会进步，所以我必须不停地前进。
C、分析派－在问题未发生之前，就该想好所有的可能。
D、无忧派－每天的生活开心快乐最重要。
4、 对于规则，我内心的态度是：
A、不愿违反规则，但可能因为松散而无法达到规则的要求。
B、打破规则，希望由自己来制定规则而不是遵守规则。
C、严格遵守规则，并且竭尽全力做到规则内的最好。
D、不喜被规则束缚，不按规则出牌会觉得新鲜有趣。
5、 我认为自己在行为上的基本特点是：
A、慢条斯理，办事按部就班，能与周围的人协调一致。
B、目标明确，集中精力为实现目标而努力，善于抓住核心要点。
C、慎重小心，为做好预防及善后，会不惜一切而尽心操劳。
D、丰富跃动，不喜欢制度和约束，倾向于快速反应。
6、 当我做错事时，我倾向于：
A、害怕但表面不露声色。
B、不承认而且辩驳，但内心其实已经明白。
C、愧疚和痛苦，容易停留在自我压抑中。
D、难为情，希望逃避别人的批评。
7、 当结束一段刻骨铭心的感情时，我会：
A、很难受，可日子总要过，时间会冲淡一切的。
B、虽然觉得受伤，但一旦下定决心，就会努力把过去的影子摔掉。
C、深陷在悲伤的情绪中，在相当长的时期里难以自拔，也不愿再接受新的人
D、痛不欲生，需要找朋友倾诉或者找到渠道发泄，寻求化解之道。
8、 面对他人的倾诉，我回顾自己大多时候本能上倾向于：
A、能够认同并理解对方当时的感受。
B、快速做出一些定论或判断。
C、给予一些分析或推理，帮助对方理顺思路。
D、可能会随着他的情绪起伏而起伏，也会发表一些评论或意见。
9、 我在以下哪个群体中交流较感满足？
A、舒服轻松的氛围中，心平气和地最终达成一致结论。
B、彼此展开充分激烈的辩论并有收获。
C、有意义地详细讨论事情的好坏和影响。
D、很开心并且随意无拘束地闲谈。
10、 在内心的真实想法里，我觉得工作：

A、不必有太大压力，可以让我做我熟悉的工作就很不错。
B、应该以最快的速度完成，且争取去完成更多的任务。
C、要么不做，要做就做到最好。
D、如果能将好玩融合其中那就太棒了，不过如果不喜欢的工作实在没劲。
11、 如果我是领导，我内心更希望在部属心目中，我是：
A、可以亲近的和善于为他们着想的。
B、有很强的能力和富有领导力的。
C、公平公正且足以信赖的。
D、被他们喜欢并且觉得富有感召力的。
12、 我对认同的需求是：
A、无论别人是否认同，生活都是要继续的。
B、精英群体的认同最重要。
C、只要我在乎的那些人认同我就足够了。
D、所见之人无论贵贱都对我认同那有多好。
13、 当我还是个孩子的时候，我：
A、不太会积极尝试新事物，通常比较喜欢旧有的和熟悉的。
B、是孩子王，大家经常听我的决定。
C、害羞见生人，有意识地回避。
D、调皮可爱，乐观而又热心。
14、 如果我是父母，我也许是：
A、容易说服或者宽容的。
B、比较严厉、性急及说一不二的。
C、坚持自己的想法和比较挑剔的。
D、积极参与到子女中一起玩，被小朋友们们热烈欢迎的。
15、以下有四组格言，哪组里整体上最符合我的感觉?
A、最深刻的真理是最简单和最平凡的。要在人世间取得成功必须大智若愚。好脾气是一个人在社交中所能穿着的最佳服饰。知足是人生在世最大的幸福。
B、走自己的路，让人家去说吧。虽然世界充满了苦难，但是苦难总是能战胜的。有所成就是人生唯一的真正的乐趣。对我而言解决一个问题和享受一个假期一样好。
C、一个不注意小事情的人，永远不会成功大事业。理性是灵魂中最高贵的因素。切忌浮夸铺张。与其说得过分，不如说得不全。谨慎比大胆要有力量得多。
D、幸福在于对生命的喜悦和激情。任何时候都要最真实地对待你自己，这比什么都重要。使生活变成幻想，再把幻想化为现实。幸福不在于拥有金钱，而在于获得成就时的喜悦以及产生创造力的激情。

3.2 英语面试

3.2.1 英语面试总体介绍

首先先祝贺各位同学成功进入科软的复试环节，复试中，英语所占的比列不是很大，在我看来，对于一志愿的考生，英语复试只要不出什么太大的纰漏就行（比如说不要全场说中文，不要影响社会和谐.....），对于调剂的考生，还是要争取把每个环节都做到最好吧。

接下来给大家说说英语复试的一些个人心得。首先建议大家先参加完专业课复试再来参加英语复试，因为专业课复试是整个复试环节的大头，万一英语复试表现不好而影响到了专业面可就不值得了！

其次，大家在准备英语面试的时候别太紧张就行。下面谈谈我和一些学长学姐总结的心得：

（1）心态：各位同学你们要明白，我在开头也讲过了，科软的英语面是基本不刷人的，尤其是一志愿的考生，你们真的不用在此太过于紧张。对于调剂的考生，英语面同样不用过于有压力，按照我接下来所说的一步步去准备，相信你们也都没什么问题！

（2）心法：科软的英语面大部分同学首先上来都是被要求做一个三到五分钟英语自我介绍。各位同学我们换位思考一下，如果你是老师，一上午听那么多的自我介绍，心态再好也会有些疲惫吧！所以，进入教室的时候一定别忘了给老师打声招呼（用英语啊！Good morning teachers^_^），然后再给一个嘴角上扬八度的微笑，先让老师开心一下嘛！试想，如果你没做而别人做了，在老师那里的感觉可能就是一个是没礼貌，一个是知书达理阳光向上的好少年。。。。。

打完招呼，就安静的坐好听老师说话。听不懂也没关系，因为百分之九十都是让你自我介绍一下，时间不要太长，三到五分中足以。好了，我们再换位思考一下，老师们一上午听了那么多同学的自我介绍肯定是有点审美听觉疲劳了（除非你长得像吴亦凡不然你那嘴角上扬八度的笑容还是不足以让老师给你打 Perfect），这个时候你要学会主动的带领老师们去听你的介绍内容。

比如说，你对篮球的话题比较熟悉，那么你在自我介绍时候就可以说你爱打篮球，在读到 basketball 时候音调要高一个八度，目的就是带领老师，让老师接下来的提问能够进入你擅长你能够掌控的范围。打个比方，马上要过情人节了，你和你女朋友逛商场，看中一双耐克球鞋，想要但又不好意思直说，这个时候轻轻滴拉着她的小手，把她拽到身边耳语道“你说我 **43 码的脚**穿上这款**白色**的球鞋在**情人节那天的晚上**拉着你走在路上会不会有一种王子和公主的感呢。。？”

（3）实操：

面试时的自我介绍主要从以下几个方面进行：

1.开场白

2.姓名，英文名，毕业院校，毕业专业，毕业学院

3.性格，爱好，实践经验

4.为什么想读研,将来愿意从事的方向,读研时的打算

5.结束语

下面给出一篇自我介绍的范文：

Good morning , I am glad to be here for this interview .

First let me introduce myself. My name is XXX, 22 Years old. I come from XiaoGan City HuBei Province . I study in China Three Gorges University .And my major is Computer Science And Technology .

I have many interests such as reading books , watch movies and play ping-pong, but I most interested in Programming , especially web programming . During my four years in university , I build many websites .In grade one , I write my first personal homepage ,which gives me great pleasure . Later I use wordpress to build my personal blog , and when I learn PHP by myself ,I used it to write my own blog program . Though I did not make any money by my site , I think it is very meaningful , because it help me to build self-confidence .

In grade two and grade three , I entered our school's QiMingXing studio, at where I learnt many useful things ,such as team spirit . I think it is important to learn from others ,and we should help each other during the project .

At the end of grade three , I decided to take the National Graduate Entrance Examination . When asked about why I want to be a graduate student ,at least I have two answers . First , I think I need to learn more . In university I learnt too much languages but did not learn deep enough even in one subject. I think if I was a graduate student I can learn something deeply and be expert in some subject . Second , I think I will have a brilliant future if I can be a graduate student , I can have good salary and make my parents happy , and my life will be more smooth .

That's all , thank you very much!

上面是一篇参考范文，大家其实可以直接背诵，但是相关内容要根据自己的实际情况进行更改。下面列举了一些**按照上面自我介绍框**

架所列出来的背诵模板，同学们可以根据自己的实际情况进行排列组合，内容不一定要很复杂很长（说不定说到一半就被老师打断提问了），核心思路就是尽量选自己擅长的部分着重去介绍，引导老师去提问你想让他们提问的问题。框架中的内容只是个思路，不用全部包扩，能够把时间控制在 3~5 分钟即可。

3.2.2 模板一

1.自我介绍(self-introduce)

Good morning. I am glad to be here for this interview. First let me introduce myself. My name is ***, 24. I come from ******,the capital of *******Province. I graduated from the ******* department of *****University in July ,2001.In the past two years I have been preparing for the postgraduate examination while I have been teaching *****in NO.****middle School and I was a head-teacher of a class in junior grade two.Now all my hard work has got a result since I have a chance to be interview by you .

I am open-minded ,quick in thought and very fond of history.In my spare time,I have broad interests like many other youngers.I like reading books, especially those about *******.Frequently I exchange with other people by making comments in the forum on line.In addition ,during my college years,I was once a Net-bar technician.So, I have a comparative good command of network application.I am able to operate the computer well.I am skillful in searching for information in Internet.I am a football fan for years.Italian team is my favorite.Anyway,I feel great pity for our country‟s team. I always believe that one will easily lag behind unless he keeps on learning .Of course, if I am given a chance to study ****** in this famous University,I will stare no effort to master a good command of advance ******

2.考研原因 (reasons for my choice)

There are several reasons.

I have been deeply impressed by the academic atmosphere when I came here last summer. In my opinion, as one of the most famous ******in our country, it provide people with enough room to get further enrichment . This is the first reason.

The second one is I am long for doing research in ******throughout my life. Its a pleasure to be with my favorite ******for lifetime. I suppose this is the most important factor in my decision. Thirdly, I learnt a lot from my *****job during the past two years. However, I think

further study is still urgent for me to realize self-value. Life is precious. It is necessary to seize any chance for self-development, especially in this competitive modern world.

In a word, I am looking forward to making a solid foundation for future profession after two years study here.

3.研究生期间的计划 (plans in the postgraduate study)

First, I hope I can form systematic view of *****. As for ******, my express wish is to get a complete comprehension of the formation and development as well as **********. If possible, I will go on with my study for doctorate degree.

In a word, I am looking forward to making a solid foundation for future profession after two years study here.

4.介绍你的家乡(about hometown)

I am from , a famous city with a long history over 2,200 years. It is called “Rong Cheng ” because there were lots of banians even 900 years ago. The city lies in the eastern part of the province. It is the center of politics, economy and culture. Many celebrities were born here, for instance, Yanfu, Xie Bingxin, Lin Zexu and so on . . You know, there is a saying that “The greatness of a man lends a glory to a place”. I think the city really deserves it. The top three art ware are Shoushan Stone, cattle-horn combs and bodiless lacquerware.

In addition, it is famous for the hot springs. They are known for high-quality. Visitors at home and abroad feel it comfortable bathing here.There is my beloved hometown.

5.你的家庭(about family)

There are four members in my family, my parents, my cute cat of 9 years old and me. My father is a technician in the Fujian TV station. He often goes out on business. So most of the housework is done by my industrious mom. Climbing at weekends is our common interest. The fresh air and natural beauty can help us get rid of tiredness. They can strengthen our relation too. During my preparing for coming here, my parents' love and support have always been my power and I hope in future I will be able to repay them.

6.你的大学(about university)

*****University is the oldest one in the province. It was founded in *****and covers an area of over***** mu. It develops into a comprehensive university with efforts of generations, especially after the reform and opening-up. It takes the lead among the *****universities with nice teaching and scientific research ability. The library has a storage of *****books. Various research institutes are set up including 52 research centers. There are teaching research experimental bases. For example, the computer center, analyzing-test center, modern education technical center and so on.

7.对英语的态度

I like English very much, I am fond of watching English films and reading English novel. I passed College English Test Band Six in December, 2003. I do believe there is still a long way for me to learn English well enough. I will not shrink back,because I realize that English is a bridge connects our country with the outside world. Learning English is the most direct and available method for intercourse among countries and also useful for us to get advanced knowledge, technology and culture from other nations.

3.2.3 模板二

1. 开场白

Good morning. I am very glad to be here for this interview.

2. 姓名，英文名，毕业院校，毕业专业，毕业学院

First let me introduce myself. My name is LiShuai, and my English name is Jacky Lee. I've finished my undergraduate education in XiDian University, Majoring in Electronic Science and Technology in the college of Technical Physics.

3. 性格，爱好，实践经验

I am open-minded, willing and have broad interests like basketball, reading and especially in engineering such as software programming, website design, hardware design. For example, during the past four years, I have accomplished two websites: one is the website of our school, and the other is the website of the doctor forum of china 2007. Furthermore, I am interested in C plus plus programming language and have written some application programs. In July in the last year,I finished my graduate

project with flying colors,which was a software application about Image Process . In addition, I have also finished some projects about embedded system by using MCU when I was a junior.

4. 为什么想读研，将来愿意从事的方向，读研时的打算

Although I have broad interests in many aspects and grasp the essential knowledge of the major, but I think at present, I can do many things in a superficial level, but not be competent to do things professionally owing to lack of ample knowledge and ability. So I think further study is still urgent for me to realize self-value.

The major that I hope pursue for my further education is IC design. Because I find integrated circuits are playing a more and more important role in our modern society. And nowadays in China, with the recognition by the government, our domestic integrated circuits industry is growing rapidly and that may provide a lot of chances to us.

I plan to concentrate on study and research in this field in my graduate time. And I hope I can form a systematic view of micro electronics and IC design technology and make a solid foundation for future profession after three years study here.

5. 结束语

OK, that's all. Thank you very much.

3.2.4 模板三

1..考研原因 (reasons for my choice)

During the past four years, I have learned a lot of professional knowledge and practical skills, but gradually, I realize it is not enough. In my opinion, further study is actually urgent for me to realize and finally achieve self-value. Life is precious, it is necessary to catch any opportunity for self-development, especially in the competitive modern society. Therefore, I prefer to go on for further education.

2. 研究生期间你的计划 (plans in the postgraduate study)

First, I hope I can form systematic view of *****. As for ******,

my express wish is to get a complete comprehension of the formation and development as well as **********. If possible, I will go on with my study for doctorate degree.

In a word, I am looking forward to making a solid foundation for future profession after two years study here.

3 .介绍你的家乡(about hometown)

同上。

4. 你的家庭(about family)

There are four members in my family; my parents, my cute cat of 9 years old and me. My father is a technician in the FuJian TV station. He often goes out on business. So most of the housework is done by my industrious mom. Climbing at weekends is our common interest. The fresh air and natural beauty can help us get rid of tiredness. They can strengthen our relation, too.During my preparing for coming here ,my parents love and support have always been my power.and I hope in future I will be able to repay them.

5. 你的大学(about university)

I studied in XX University. Although it is not well know, I still appreciate it, because it offers me a chance to develop my abilities. During my college years, I have made rapid and great progress in many areas, as a student, I work very hard, and obtain scholarship many times, as a monitor, I work earnestly, also gain good comments from teachers and classmates, working as a member of Student Union, I strive to finish any assignment perfectly. In a word, I learned a lot in my college life.

3.2.5 模板四

1.研究生计划

If luckily i got the chance to learn xxx in USTC, i will concentrate on the study and research in this field. first i will hard to learn the theoretical knowledge,constructing a solid base for my future work; second I would like to do some practical work with the help of the supervisor classmate .and through this ,i can get something that cannot be acquired from the textbooks. i believe after 2 years of learning ,my dream will finally come true..

2.自我介绍:

Good morning, everyone! I am glad to be here for this interview. First, let me introduce myself to you. My name is XXX. I was born on XX, 19XX. I am a local person.I am graduating from Jilin Normal University this June. I major in Chinese literature. I hope I could get the opportunity to finish my postgraduate courses in Jilin University which I have desired for a long time. I have the confidence because I have such ability! I am a girl who is fervent, outgoing and creative. At the same time, I think I am quick in mind and careful in everything. I am looking forward to my postgraduate studies and life. I will soon prove that your decision of choosing me is the wisest. Thank you for giving me such a valuable opportunity!

3.考研原因:

First of all, I love my major. Chinese literature is the symbol of the start of Chinese literary modernization. It plays an important part in modernization of our citizens' thoughts. What's more, modern literature is very close to our daily life and it can deeply reflect the styles and features of our society. I am fascinated by the great masters' refreshing or warm or profound styles as well. But I am not easily satisfied with such superficial knowledge. I hope I could have a better understanding in modern literature by studying further. This is a very important reason for me to take the postgraduate exams.

Next, I love the feeling in the university. It is full of youthful spirit. And I am deeply attracted by the scholarly atmosphere. And the most important, it's my great honor to open my ears to your teaching.

Finally, I want to talk about a very practical problem. That is my dream of becoming a teacher in the university. I want to realize my dream and make myself to be a well-qualified person. I think the postgraduate studies can enrich my knowledge and make me competent in my future job.

That's my simple and clear reasons why I took the postgraduate exams.

4.介绍我的大学:

I'm graduating form Jilin Normal University this June which has a history of 50 years. It shares many same characteristics with Jilin University. Both of them have a refreshing and scholarly atmosphere.

Four years' studying there made me an independent, optimistic and strict girl. I appreciate the education my university gave me.

3.2.6 总结：

上面的模版只是用来作为参考，给学弟学妹们的思路上一些启发。考虑到科大软院复试人数众多，为了防止背诵雷同，同时所选的模版也不一定适合各位的实际情况，在此，我进一步建议大家仅以此模版作为参考，适当排列组合好模版的内容，当然同学们最好可以针对自己的实际情况写出一份更适合自身的模版，比如可以先用汉语写出来再结合翻译软件。

最后，想必各位同学看到此文的时候也快过年了吧，在此提前祝各位同学新年快乐，把梦想都能全部实现吧！

3.3 专业课面试

3.3.1 数据结构

这里对复试的专业课部分进行简要说明。我本人去年一志愿报考的软院，因为是跨考，复试前说实话也挺紧张，虽然自己初试成绩还算理想，但就怕自己复试的时候一问三不知，要是因为复试被刷那可就真是太可惜了。我去年专业课复试的时候对面坐了三个老师，老师人都很好，貌似有科大西区六系那边的。

进来以后跟老师打完招呼就赶紧坐下来了，老师先是让我自我介绍下，主要就说说自己之前的学校专业以及为什么选择了科软吧。可能是因为我跨的专业还比较大，几个老师对我多少还是有了些些疑虑，记得当时有个老师就直说"你这个专业跨度有点太大了吧。。。。。"但后来想想跨专业也有跨考的好处，就是老师在随机提问的过程中问的问题都比较简单哈哈哈啊。

简单了解我的情况后，老师就让我开始自己抽题（主要就是 834 考的那些，貌似还有些数据库，编译原理啥的）。这里大家还是选择自己比较擅长的科目吧，自己寒假玩的比较疯，也就只复习了数据结构的部分，所以想都没想就翻了数据结构的牌。（我们那一组是让我们自己抽签选提问科目的，但有的组没有这一环节，老师上来就是一顿怼）抽的题目相对还比较简单，我也在后面的列题中的前两个给大家回顾了一下。

这里告诉大家一个小技巧就是一定要多问问在你之前专业面的同学啊，我被抽到的题和我之前的一个同学真是一毛一样，幸好在等

待复试的时候提前百度了一下，不然可能真的答不出来。

答完抽签的题后，几位老师又看了看我的成绩单，看我学过电工电子技术，于是问了我二极管和三极管的区别是啥，工作原理又是啥。。。。。。在这里提醒下调剂的考生，据我所知，去年大部分调剂的考生好像就没有抽题这一环节了，老师都是看你成绩单问问你本科的相关科目，所以调剂的考生的确不好针对性的准备专业面，不过还是建议翻翻数据结构的相关概念外加把本科和计算机，电子等相关的内容再瞅瞅吧！

下面列举了些科软复试的时候数据结构部分可能被问到的题型，其中前两题是本人去年复试的时候抽到的原题；

1. 图的相关概念（）

图：由结点的有穷集合 V 和边的集合 E 组成。

类别：有向图和无向图。

顶点的度：出度和入度。

有向完全图和无向完全图：**若有向图有 n 个顶点，则最多有 n(n-1) 条边，则称为有向完全图；若无向图有 n 个顶点，则最多有 n(n-1)/2 条边，则称为无向完全图。**

路径：相邻顶点序偶所构成的序列。

简单路径：序列中的顶点和路径不重复出现的路径。

回路：路径中第一个顶点和最后一个顶点相同的路径。

连通：无向图中，如果 Vi 到 Vj 有路径，则称这两个顶点连通。如果图中任意两个顶点之间都连通，则称改图为连通图。

有向图中，如果 Vi 到 Vj 有路径，则称这两个顶点连通。如果图中每一对顶点 Vi 和 Vj，从 Vi 到 Vj 和 Vj 到 Vi 都有路径，则称改图为强连通图。

2. c 语言相关

我们知道在 C 语言中如果要交换两个变量的值通常需要引入第三个变量 t,如 a=1,b=2；交换 a,b 的值可以采用的语句为 int t;t=a;a=b;b=t;那么现在如何能在不引入多余变量 t 的条件下就能够交换 a,b 的值呢？

（tip:这道题去年复试的时候至少有两三个同学包括我抽到，复试的时候会分成很多组，如果你的分组比较靠后的话一定要多问问之前复试结束的同学他们抽到了哪些题，此题就是我了解到之前有人抽到了该题目，于是我就百度搜了一下，没想到。。。我居然也抽到了哈哈）

3. 时间复杂度

时间复杂度是指执行算法所需要的计算工作量，因为整个算法的执行时间与基本操作重复执行的次数成正比，所以将算法中基本操作的次数作为算法时间复杂度的度量，一般情况下，按照基本操作次数最多的输入来计算时间复杂度，并且多数情况下我们去最深层循环内的语句所描述的操作作为基本操作。

4. 单链表中，增加一个附加头节点的目的是为了？

方便运算的实现，增加附加信息，标志单链表存在。

5. 什么是二叉排序树？以及它的原理，算法。（二叉排序树的查找过程）

二叉排序树又称二叉查找树，它或者是一颗空树，或者满足一下性质的二叉树：

① 若左子树不空，则左子树上所有结点的值均小于根节点的值；

② 若右子树不空，则右子树上所有结点的值均大于根节点的值；

③ 左右子树也分别是二叉排序树。

原理步骤：

若根结点的关键字值等于查找的关键字，成功。

否则，若小于根结点的关键字值，递归查左子树。

若大于根结点的关键字值，递归查右子树。

若子树为空，查找不成功。

6. 含有 N 个节点的二叉树共有多少种？

$$\frac{c_{2n}^{n}}{n+1}$$

7. 二叉树和树的主要不同有哪些？

（1）树中结点的度多少没有限制，而二叉树结点度最大为 2；

（2）二叉树节点有左右之分，而树无；

8. 无向图和有向图的区别

（1）无向图是对称的，主对角线一定为 0，有向图则**不一定**；

（2）度：无向图中，第 i 行/列或的元素之和即为 i 的度；

有向图中，第 i 行（j 列）元素之和为点 i（j）的出度（入度）

（3）边数：无向图中边数是矩阵中非零元素个数的一半；有向图中边数是非零元素的个数；
（4）存储空间：有向图为 n^2,无向图为 n(n-1)/2。

9. 什么是哈夫曼树

定义：

给定 n 个权值作为 n 个叶子结点，构造一棵二叉树，若带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman tree)。

构造方法：

假设有 n 个权值，则构造出的哈夫曼树有 n 个叶子结点。n 个权值分别设为 w1、w2、…、wn，则哈夫曼树的构造规则为：

(1) 将 w1、w2、…，wn 看成是有 n 棵树的森林(每棵树仅有一个结点)；

(2) 在森林中选出两个根结点的权值最小的树合并，作为一棵新树的左、右子树，且新树的根结点权值为其左、右子树根结点权值之和；

(3)从森林中删除选取的两棵树，并将新树加入森林；

(4)重复(2)、(3)步，直到森林中只剩一棵树为止，该树即为所求得的哈夫曼树。

特点：

① 权值越大的结点，距离根节点越近；

② 树中没有度为一的结点。

应用：

哈夫曼编码，减少编码的长度。哈夫曼编码就是长度最短的前缀编码。

10. 既希望较快的搜索，又便于线性表动态变化的搜素方法是？

索引顺序查找：该方法索引块内部的数据可以无序，故在块内方便进行数据的删除或增加。

11. 什么是哈希冲突？以及如何解决？

散列（哈希）表：

根据关键码值(Key value)而直接进行访问的数据结构。根据给定的关键字来计算出关键字在表中的地址，以加快查找的速度。

冲突：指的是多个关键字映射同一个地址的情况。

解决办法：

（1）开放定址法

① 线性探查法（产生堆积问题）；

② 平方探查法（不能探查到哈希表上所有的地址，但至少能探查到一半的地址）

（2）链地址法

把所有的同义词用单链表连接起来。

补充（常见的哈希函数构造方法）

直接定址法，数字分析法，平方取中法，除留余数法。

12. 深度优先搜索遍历和广度优先搜索遍历的过程

深度优先搜索遍历

基本思想：首先访问出发点 V，并将其标记为已访问；然后选取与 V 邻接的未被访问的邻接顶点 W，访问 W；再选取与 W 邻接的未被访问的顶点访问，以此类推。当一个顶点所有的邻接顶点都被访问过时，则依次退回最近被访问过的顶点，若该顶点还有其他邻接顶点未被访问，则从这些顶点中去一个顶点进行上述的过程，直至图中所有顶点都被访问过为止。

广度优先搜索遍历

基本思想：首先访问起始顶点 V，然后选取与 V 邻接的全部顶点 w1，w2，…，wn 进行访问，再一次访问与 w1，w2，…，wn 邻接的全部顶点（不包括已访问过的顶点），以此类推，直至所有顶点都被访问过为止。

13. 链表查找某个元素，平均的时间复杂度是多少？

O(n)　链表是顺序存储，故（1+n）/2。

14. 图的深度遍历是否唯一

不一定是唯一的。当图是树的时候唯一，否则不唯一。

15. 最小生成树的概念？

一个有 n 个结点的连通图的生成树是原图的极小连通子图，且包含原图中的所有 n 个结点，并且有保持图联通的最少的边。如果在最小生成树中添加一条边，必定成一个环。

相关算法：

① 普里姆算法

② 克鲁斯卡尔算法

N 个结点的最小生成树有几个结点，几条边：n 个结点，n-1 条边。

16. 平衡二叉树

平衡二叉树又称 AVL 树，是一种特殊的二叉排序树，其左右子树都是平衡二叉树，且左右子树的高度差的绝对值不超过 1.

平衡因子： 左子树高度减去右子树高度的差。

平衡调整： 先找到失去平衡的最小子树，即以距离插入结点最近，且平衡因子绝对值大于 1 的结点最为根节点的子树，分为 LL,LR,RL,RR 四中调节方式。

17. 折半查找，以及其适用范围和时间复杂度？

又称二分查找，

基本思路：

在当前的查找区间[low…high]中，首先确定 mid=(low+high)/2，然后拿关键字与 mid 比较，若相等则查找成功，返回该位置，否则确定新的查找区间， mid>K，[low…mid-1]

mid<K，[mid+1…high]

直至查找自区间长度小于 1 时查找结束。

适用范围：顺序结构存储并按照关键字大小有序排列。

时间复杂度：O(log2N)

18. 什么是完全二叉树？

若一棵二叉树至多只有最下面的两层上的结点的度数可以小于 2，并且最下层上的结点都集中在该层最左边的若干位置上，则此二叉树成为完全二叉树。

完全二叉树特点：

叶子结点只可能在最大的两层上出现， 对任意结点， 若其右分支下的子孙最大层次为 L，则其左分支下的子孙的最大层次必为 L 或 L+1；

19. 什么是堆？有什么作用？

堆是一种数据结构，可以把堆看成一个完全二叉树，并且这个完全二叉树满足：

任何一个非叶节点的值都不大于（或不小于）其左右子树的结点的值。若父亲大孩子小，则为大顶堆，若父亲肖孩子大，则为小顶堆。

作用：应用于堆排序。

20. 如何实现循环队列？有何好处？

如何实现：把数组弄成一个环，让 rear 和 front 指针沿着环走，这样就可以产生循环队列。

好处：循环队列是顺序队列的改进，在顺序队列中，在元素进队

的时候，rear 要向后移动，元素出队的时候，front 也要向后移动，这样经过一系列的出队和入队操作之后，两个指针最后会达到数组的末端，此时虽然队中已经没有元素了，但是还是不能让元素入队，即出现了“假溢出”的现象。循环队列就能避免出现这个现象。

21. 深度优先搜索形成的是什么？森林唯一么？

（森林，不能说树）（不唯一，因为邻接表可能不唯一）

22. 满二叉树的结点个数（n 层）

2 的 n 次方减一（2^n-1）

23. 什么图可以进行拓扑排序？

有向无环图

24. 连算法的时间复杂度取决于？

（1）问题的规模；

（2）待处理数据的初态。

25. 线性表的静态链表存储结构与顺序存储结构相比，其优点是？

便于插入和删除

26. 需要分配较大的空间，插入和删除不需要移动元素的线性表，其存储结构为？

静态链表

27. 栈和队列的共同点是？

只允许在端点处插入和删除元素；限制存取点的线性结构；

28. 什么是关键路径？

事件结点图中从源点到汇点的最长路径。

29. 查找效率最高的二叉排序树为？

所有结点的右子树都为空的二叉排序树

30. 设有 n 个关键字，hash（哈希）查找法的时间复杂度是？

O(1)

31. 堆和栈的区别？请简要说明。

从程序的内存分配来看：

（1）栈区（stack）— 由编译器自动分配释放，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。

（2）堆区（heap）— 般由程序员分配释放， 若程序员不释放，程序结束时可能由 OS 回收 。注意它与数据结构中的堆是两回事，分配方式倒是类似于链表。

（3）全局区（静态区）（static）— 全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域， 未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。 程序结束后由系统释放。

（4）文字常量区 — 常量字符串就是放在这里的。程序结束后由系统释放。

（5）程序代码区 — 存放函数体的二进制代码。

堆和栈的区别可以用如下的比喻来看出：

使用栈就象我们去饭馆里吃饭，只管点菜（发出申请）、付钱、和吃（使用），吃饱了就走，不必理会切菜、洗菜等准备工作和洗碗、刷锅等扫尾工作，他的好处是快捷，但是自由度小。使用堆就象是自己动手做喜欢吃的菜肴，比较麻烦，但是比较符合自己的口味，而且自由度大。

32. M 阶 B-树和 M 阶 B+树的主要区别

① B+树所有有效数据全在叶子节点，而 B-树所有节点分散在树中，B-树中的关键字不重复。

② B+树种有几个关键字就有几个子树，B-树中具有 n 个关键字的节点含有(n+1)棵子树。

③ B+树有两个指针，根指针和只想最小节点的指针，叶子节点连接成一个不定长的线性链表

④ B+树中，每个节点（除根节点外）中的关键字个数 n 的取值范围是 m/2 <=n<=m，根节点 n 的取值

⑤ 范围是 2<=n<=m。B-树中，每个节点（除根节点外的所有最底层非叶子节点）中的关键字取值范围是 m/2 -1<=n<=m-1，根节点 n 的取值范围是 1<=n<[m-1]。

⑥ B+树中的所有非叶子节点仅仅起到索引的作用，节点中的每个索引项只包含对应子树的最大关键字和指向该子树的指针，不含有该关键字对应记录的存储地址。而在 B-树中，每个关键字对应记录的存储地址。

33. 迪杰斯特拉算法的过程、克鲁斯卡尔算法的过程、普锐姆算法的过程

该算法可以求得某一顶点到其余各顶点的最短路径。
算法思想：设有两个顶点集合 S 和 T，其中集合 S 中存放的是图中已找到最短路径的顶点，集合 T 中存放的是图中的剩余顶点。
初始状态时，集合 S 中只包含源点 V0，然后不断从集合 T 中选取到顶点 V0 路径最短的顶点 Vu 并加入集合 S 中。集合 S 每加入一个新的顶点 Vu，都要修改 V0 到集合 T 中各个顶点的最短路径的长度值。不断重复这个过程，直至集合 T 中的顶点全部并入到 S 中为止。

34. 图的存储方式

① 邻接矩阵：是图的顺序存储结构，用两个数组分别存储数据元素（顶点）信息和数据元素之间的关系（边/弧）的信息。图的邻接矩阵表示是唯一的，无向图的邻接矩阵是对称的。
② 邻接表：是图的链式存储结构，由单链表的表头形成的顶点表和单链表其余结点所形成的边表两部分组成。
③ 十字链表：有向图的另一种链式存储结构。
④ 邻接多重表：无向图的链式存储结构。

3.3.2 微机原理

1、微机系统的基本组成是什么呢？

简单的说，它由硬件系统和软件系统两部分组成，并且采用总线结构。

首先，**硬件系统**构成了微机系统的全部物理装置。它由 5 部分组成：

1） 存储器：存放数据和程序
2） 微处理器(CPU)：包括运算器和控制器，运算器完成二进制编码的算术和逻辑运算；控制器是控制计算机进行各种操作的部件。
3） 输入设备及其接口电路：用来输入数据、程序、命令和各种信号。如键盘，鼠标。
4） 输出设备及其接口电路：输出计算机处理的结果，如打印机，显示器。
5） 网络设备。

其次，**软件系统**是指计算机所编制的各种程序的集合，可分为系统软件和应用软件两大类。

1） **系统软件**是用于实现对计算机资源管理、控制和维护，方便人们使用计算机而配置的软件，如操作系统，语言的汇编、解释、编译程序，数据库管理程序。

2） **应用软件**是指用户利用计算机以及系统软件编制的解决实际问题的程序，它包括支撑软件和用户自己编写的程序。支撑软件包括 WPS、AutoCAD、Microsoft-Office 等。

硬件系统是人们操作微机的物理基础；软件系统是什么与微机系统进行信息交换、通信对话、以人的思维控制和管理微机系统的工具。

2、微机的总线如何分类的？

从总线的不同使用层次可以分为：

1） **内部总线：**微处理器内部各个部件之间传送信息的通路。它是由微处理器芯片厂家生产设计的。

2） **元件级总线：** 连接计算机系统中两个主要部件的总线。包括地址总线、数据总线和控制总线。

地址总线是 CPU 用来向存储器或 I/O 端口传送地址的，是三态单向总线。

数据总线是 CPU 与存储器及外设交换数据的通路，是三态双向总线。

控制总线是用来传输控制信号的，传送方向就是由它而定，如 CPU 向存储器或 I/O 接口电路输出读信号、写信号、地址有效信号，而 I/O 接口部件向 CPU 输入复位信号、中断请求信号和总线请求信号等。

3） **系统总线：**微处理机机箱内的底板总线，用来连接构成微处理机的各个插件板。主要有 ISA、EISA、VESA、PCI 几类总线。

4） **外部总线：**用于微处理机系统与系统之间、系统与外部设备之间的信息通路。这种总线数据的传送方式有并行和串行方式。例如 USB 总线就是通用串行总线。

3、计算机的数据格式是什么？有哪些表现形式？

计算机中的数用二进制表示，数的符号也用二进制表示，最高有效位代表符号，0 表示正数，1 表示负数，将一个数与符号用数值化表示，称为机器数。

机器数的字长由计算机字长决定，即决定了机器数的表示范围。如 8 位字长的微机可表示 256 个数，对于无符号数，取数范围为 0~255；对于有符号数，取数范围为-128~+127.

1）机器数可以用原码，反码和补码表示，常用的是补码表示法。

2）二进制编码的十进制（BCD）是将十进制数的每一位以二进制数编码方式表示，十进制的 0~9 分别用 BCD 数的 0000~1001 表示，而不是整个十进制数转换成二进制形式。BCD 码有压缩和非压缩

两种形式。压缩 BCD 数据以每字节 2 个数字的形式存储，非压缩 BCD 数据以每字节 1 个数字的形式存储。常用的是压缩 BCD 码。

3）**ASCII 码：**从键盘输入的信息（数字和字符）或显示输出的信息都是以字符方式输入输出的，这些字符通常用美国标准信息交换代码 ASCII 码来表示，ASCII 码用 8 位二进制码表示一个字符。可表示 128 个字符，其中可打印字符有 96 个（包括空格），另外有 32 个控制字符。

4、引入补码的原因是什么？

1）符号位和有效数值部分一起参加数值运算，简化运算规则，节省运行时间。

2)使得减法运算转化为加法运算，简化计算机中运算器的线路设计。

5、8086CPU 的组成及部件功能是什么？

8086CPU 由总线接口部件 BIU 和指令执行部件 EU 组成，两者之间相互配合又相互独立的非同步并行工作方式提高了 CPU 的工作效率。

总线接口部件 BIU 是 8086CPU 与外部（存储器和 I/O 端口）的接口，它提供了 16 位双向数据总线和 20 位地址总线，完成所有外部总线操作。

BIU：地址形成，取指令，指令排队，读/写操作数，送结果和总线控制，所有与外部的操作由其完成。

指令执行部件 EU 完成指令译码和执行指令的工作。

EU：从 BIU 指令队列中取指令，执行指令，不必访问存储器或 I/O 端口。

若需要访问存储器或 I/O 端口，也是由 EU 向 BIU 发出发出访问所需地址，在 BIU 中形成物理地址，然后访问存储器或 I/O 端口，取得操作数送到 EU，或送结果到指定的内存单元或 I/O 端口。

6、BIU 和 EU 的组成：

BIU：1）16 位段地址寄存器：CS,DS,ES,SS

2) 16 位指令指针寄存器 IP：存放下一条要执行指令的偏移地址

3）20 位物理地址加法器：将 16 位逻辑地址变换成存储器读/写所需的 20 位物理地址，实际完成地址加法操作。

4）6 字节指令队列：存放 6 字节的指令代码

5）总线控制逻辑：发出总线控制信号

EU: 1）算术逻辑运算单元 ALU：完成 8 位/16 位的二进制运算，16 位暂存器可暂存参加运算的操作数。

2) 标志寄存器 flags：存放 ALU 运算结果特征
3）寄存器组：4 个通用 16 位寄存器累加器 AX、基址寄存器 BX、计数寄存器 CX、数据寄存器 DX，4 个专用 16 位寄存器，源变址寄存器 SI、目的变址寄存器 DI、堆栈指针寄存器 SP、基址指针寄存器 BP。
4）EU 控制器：取指令控制和时序控制部件。

7、通用寄存器组和专用寄存器组的使用

每个通用寄存器可存放 16 位操作数，也可拆成两个 8 位寄存器，用来存放 8 位操作数。而专用寄存器可用来存放数据和地址，但只能按 16 位进行存取操作。
它们的特定用法如下：

寄 存 器	数 据	寄 存 器	数 据
AX	字乘、字除、字 I/O	CL	多位移位和循环移位
AL	字节乘、字节除、字节 I/O 查表转换、十进制运算	DX	间接 I/O 地址
AH	字节乘、字节除	SP	堆栈操作
BX	查表转换	SI	数据串操作
CX	数据串操作、循环	DI	数据串操作

8、标志寄存器和标志位：

16 位标志寄存器 flags 用来存放运算结果的特征。特征（标志位）常用作后续的条件转移指令的转移控制条件，其中 7 位没有用，其余 9 位分成两类：一类为状态标志，表示运算后结果的状态特征，影响后面的操作。另外一类为控制标志，控制 CPU 操作。状态标志有 CF,PF,AF,ZF,SF,OF 6 个。控制标志有 3 个，即：TF、IF 和 DF。
6 个状态标志位：
1） CF：进位标志位，最高位有进位或借位，CF=1.STC 使 CF 置位，CLC 使之复位，CMC 使之取反。
2） PF：奇偶校验标志位，低 8 位中有偶数个“1”，PF=1。有奇数个 1 时，PF=0.
3） AF:辅助进位标志位，低 4 位向高 4 位有进位/借位时，AF=1.
4） ZF：全零标志位，运算结果为 0 时，ZF=1,否则 ZF=0.
5） SF：符号标志位，最高位为 1，SF=1,否则 SF=0.反映结果的正负。
6） OF：溢出标志位：产生溢出时，OF=1.
判断溢出标志：如果最高有效位和次高有效位同时有进位（借位）或同时无进位（借位）时，表示无溢出，否则表示有溢出。

判断奇偶标志：无论是 8 位还是 16 位数据，只考查其中的低 8 位数据的“1”的个数的奇偶性。

3 个**控制标志位：**

1）TF: 单步标志位。用户调试程序时，可设置单步工作方式，TF=1 时，CPU 每执行完一条指令，就会自动产生一次内部中断，使用户能够逐条跟踪程序进行调试。

2）IF：中断标志位。IF=1 时，允许 CPU 响应可屏蔽中断，当 IF=0 时，即使外部设备申请中断，CPU 也不响应。STI 使 IF 置 1，CLI 使 IF 置 0.

3）DF：方向标志位。控制串操作指令中地址指针的变化方向，在串指令中，若 DF=0，地址指针自动增量，由低地址向高地址进行串操作。DF=1 时，地址指针自动减量，由高地址向低地址进行串操作。STD 使 DF 置 1，CLD 使 DF 置 0.

9、8086/8088CPU 的两种工作模式：

最小模式：用于单机系统，系统中只有 8086/8088 一个微处理器。系统中所需的控制信号全部由 8086 直接提供。

最大模式：用于多处理机系统，系统中通常含有两个或多个微处理器。其中一个主处理器是 8086/8088CPU,另外的处理器可以是协处理器、I/O 处理器。系统中所需的控制信号由总线控制器 8288 提供。

10、8086 与 8088CPU 的不同之处：

1）8088CPU 的内部数据总线宽度是 16 位，外部数据总线宽度是 8 位，所以 8088CPU 称为准 16 位微处理器。而 8086CPU 是 16 位微处理器。

2）8088 的指令队列长度是 4 个字节，指令队列中只要出现一个空闲字节时，CPU 就会自动访问存储器，取指令来补充指令队列。（8086 要在指令队列中至少出现 2 个空闲字节时，才预取后续指令）。

3）8088CPU 中，BIU 的总线控制电路与外部交换数据的总线宽度是 8 位，总线控制电路与专用寄存器组之间的数据总线宽度也是 8 位，而 EU 的内部总线是 16 位，这样，对 16 位数的存储器读/写操作需要两个读/写周期才可以完成.

4) 8088 外部数据总线只有 8 条，所以分时复用的地址/数据总线为 AD_7~AD_0;而 AD_{15}~AD_8 成为仅传递地址信息的 A_{15}~A_8。

5）8088 中，用 IO/$\overline{M}$信号代替 M/$\overline{IO}$信号，IO/$\overline{M}$低电平时选通存储器，高电平时选通 I/O 接口，此举是为了与 8085 总线结构兼容。

6) 8088 中，只能进行 8 位数据传输，$\overline{BHE}$信号不需要了，改为 $\overline{SS_0}$，

与 DT/$\overline{R}$和 IO/$\overline{M}$一起决定最小模式中的总线周期操作。

11、8086 系统的物理地址形成：

8086 系统将段地址放在段寄存器中，称为“段基址”。段内偏移地址为从段地址开始的相对偏移位置。它可以放在指令指针寄存器 IP 或者 16 位通用寄存器中。

物理地址是存储器的绝对地址，是 CPU 访问存储器的实际寻址地址，它由逻辑地址变换而来。

逻辑地址由段基址和偏移地址组成，都是无符号的 16 位二进制数，程序设计时采用逻辑地址。由于访问存储器的操作类型不同，BIU 使用的逻辑地址来源也不同。如取指令时，自动选择 CS 寄存器值作为段基址，偏移地址由 IP 来指定。

物理地址=段基址*16+偏移地址

12、奇偶存储器的访问

8086 系统中，1MB 的存储空间分成两个存储体：**偶地址存储体**和**奇地址存储体**。各为 512KB。

当 A_0=0 时，选择访问偶地址存储体，偶地址存储体与数据总线低 8 位相连，从低 8 位数据总线读/写一个字节。

当$\overline{BHE}$=0 时，选择访问奇地址存储体，奇地址存储体与数据总线高 8 位相连，由高 8 位数据总线读/写一个字节。

当 A_0=0，$\overline{BHE}$=0 时，访问两个存储体，读/写一个字。

A_0 和$\overline{BHE}$的功能组合如图所示：

操　作	$\overline{BHE}$	AD_0	使用的数据引脚
读或写偶地址的一个字	0	0	AD_{15}～AD_0
读或写偶地址的一个字节	1	0	AD_7～AD_0
读或写奇地址的一个字节	0	1	AD_{15}～AD_0
读或写奇地址的一个字	0	1	AD_{15}～AD_8(第一个总线周期放低位数据字节)
	1	0	AD_7～AD_0(第二个总线周期放高位数据字节)

若**读/写一个字节**，则只启动一个存储体，只有相应的 8 位数据在数据总线上有效。即启动偶地址存储体，低 8 位数据线有效，或启动奇地址存储体，高 8 位数据线有效，另外 8 位数据被忽略。

若**读/写一个字**，如果字单元地址从偶地址开始，则只需要访问一次存储器，低位字节在偶地址单元，高位字节在奇地址单元。如果字单元地址从奇地址开始，则 CPU 需要两次访问存储器，第一次取奇地址上数据（偶地址 8 位数据被忽略），第二次取偶地址上数据（奇地

址 8 位数据被忽略)。因此，为了加快程序运行速度，编程时注意从存储器偶地址开始存放字数据，即“**对准存放**”。

13、堆栈的工作方式什么？它有哪些用途？

堆栈是存储器中的一个区域，用于存放需要暂时保存的数据。段基址是由堆栈寄存器 SS 指定，栈顶由堆栈指针 SP 指定，SP 一般指向当前栈顶单元。堆栈的地址增长方式一般是向上增长，栈底设在存储器的高地址区，堆栈地址由高到低增长。堆栈以字为单位进行操作，堆栈中的数据项以低字节在偶地址，高字节在奇地址的次序存放。

堆栈的工作方式是“**先进后出**”，用入栈指令 PUSH 和出栈指令 POP 可将数据压入堆栈或从堆栈中弹出数据，栈顶指针 SP 的变化由 CPU 自动管理。当执行 PUSH 指令时，CPU 自动修改指针 SP-2->SP。使 SP 指向新栈顶，然后将低位数据压入 SP 单元，高位数据压入 SP+1 单元。当执行 POP 指令时，CPU 先将当前栈顶 SP（低位数据）和 SP+1（高位数据）中的内容弹出，然后再自动修改指针 SP+2->SP，使 SP 指向新栈顶.

堆栈主要用于中断及子程序调用，也可用于数据暂时保存。在进入中断服务子程序和子程序调用前，原来 CPU 中现行信息（指令指针 IP，及寄存器中相关内容）都必须保存，在中断服务子程序和子程序调用结束返回主程序时，又必须恢复原来保存的信息，这些均由堆栈操作来完成。

由于先进后出的特点，堆栈操作应注意：

1） 先进入的内容要后弹出，保证返回寄存器内容不发生错误。

2） PUSH 和 POP 的指令要成对，如果不匹配，则会造成返回主程序的地址出错。

14、什么是指令，时钟，总线周期，它们之间有什么关系?

指令周期：执行一条指令所需的时间。不同指令的指令周期长短是不同的，一个指令周期由几个总线周期组成。

时钟周期：CPU 的时钟频率的倒数，也称为 T 状态。它是微处理器的最小动作单位时间。

总线周期：8086CPU 中，BIU 完成一次访问存储器或 I/O 端口操作所需要的时间。每个总线周期至少包含 4 个时钟周期（T_1~T_4）,一般情况下，在总线周期的 T_1 状态传送地址，T_2~T_4 状态传送数据。

15、8086 的寻址方式有哪些？

计算机指令通常包含操作码和操作数两部分，前者指出操作的性质，后者给出操作的对象。寻址方式就是指令中说明操作数所在地址的方

法。
指令有单操作数、双操作数和无操作数之分。如果是双操作数指令，要用逗号将两个操作数分开，逗号右边的操作数称为源操作数，左边的为目的操作数。
寻址方式类别:
1)立即寻址方式：操作数直接包含在指令中，它是一个 8 位或 16 位的常数，也叫立即数。这类指令翻译成机器码时，立即数作为指令的一部分，紧跟在操作码之后，存放在代码段内。如果立即数是 16 位数，则高字节存放在代码段的高地址单元中，低字节放在低地址单元中。立即寻址方式的指令常用来给寄存器赋初值。
2）寄存器寻址方式：操作数包含在寄存器中，由指令指定寄存器的名称。对于 16 位操作数，寄存器可以是 AX、BX、CX、DX、SI、DI、SP 和 BP 等。对于 8 位操作数，则用寄存器 AH、AL、BH、BL、CH、CL、DH 和 DL。
3）直接寻址方式：操作数的偏移地址称为有效地址 EA。使用直接寻址方式的指令时，存储单元的有效地址直接由指令给出，在它们的机器码中，有效地址存放在代码段中指令的操作码之后。而该地址单元中的数据总是存放在存储器中，所以必须先求出操作数的物理地址，然后再访问存储器，才能取得操作数。
4）寄存器间接寻址方式：指令中给出的寄存器中的值不是操作数本身，而是操作数的有效地址。寄存器名称外面必须加方括号，以与寄存器寻址方式相区别。这类指令中使用的寄存器有基址寄存器 BX、BP 及变址寄存器 SI、DI。
5）寄存器相对寻址方式：操作数的有效地址是一个基址或变址寄存器的内容与指令中指定的 8 位或 16 位位移量之和。这种寻址方式与寄存器间接寻址十分相似，主要区别是前者在有效地址上还要加一个位移量。
6）基址变址寻址方式的操作数的有效地址是一个基址寄存器(BX 或 BP)和一个变址寄存器(SI 或 DI)的内容之和，两个寄存器均由指令指定。
7）**相对基址变址寻址方式**的操作数的有效地址是一个基址寄存器和一个变址寄存器的内容，再加上指令中指定的 8 位或 16 位位移量之和。
8）**隐含寻址：**指令中不指明操作数，但有隐含规定的寻址方式。例如指令 DAA，它的含义是对寄存器 AL 中的数据进行十进制数调整，结果仍保留在 AL 中。
9）**I/O 端口寻址：**8086 有直接端口和间接端口两种寻址方式。在直接端口寻址方式中，端口地址由指令直接提供，它是一个 8 位立即数。

由于一个 8 位二进制数的最大值为 2^8 —1 = 255,所以在这种寻址方式中，能访问的端口号为 00～FFH，即 256 个端口。
在间接寻址方式中，被寻址的端口号由寄存器 DX 提供，这种寻址方式能访问多达 2^{16}= 64K 个 I/O 端口，端口号为 0000～FFFFH。
10）**转移类指令寻址**

16、操作数的来源及对应寻址方式的指令执行速度有什么差异？
操作数的来源有 4 种：
立即数：参与操作的数据直接在指令中。
寄存器操作数：参与操作的数据存放在寄存器中。
存储器操作数：参与操作的数据存放在存储器中。
I/O 操作数：参与操作的数据来源于 I/O 端口
操作数在寄存器中的指令执行速度最快，因为它们可以在 CPU 内部立即执行。立即数寻址指令可直接从指令队列中取数，所以它们的执行速度也较快。而操作数在存储器中的指令执行速度较慢，因为它要通过总线与 CPU 之间交换数据，当 CPU 进行读写存储器的操作时，必须先把一个偏移量送到 BIU，计算出 20 位物理地址，再执行总线周期去存取操作数，这种寻址方式最为复杂。

17、算数移位和逻辑移位指令的异同点：
指令可对寄存器或存储器中的字或字节的各位进行算术移位或逻辑移位，移动的次数由指令中的计数值决定。
SAL 算数左移指令和 SHL 逻辑左移指令的功能完全相同，均将寄存器或存储器中的目的操作数的各位左移，每移位一次，最低有效位 LSB 补 0，而最高有效位 MSB 进人标志位 CF。移位一次，相当于将目的操作数乘以 2。指令中的计数值决定所要移位的次数。若只需要移位一次，可直接将指令中的计数值置 1。若移位次数大于 1，应先将移位次数送进 CL 寄存器，再把 CL 放在指令的计数值位置上。
SHR 逻辑右移指令对目的操作数中各位进行右移，每执行一次移位操作，操作数右移一位，最低位进入 CF，最高位补 0。右移次数由计数值决定，同 SAL/SHL 指令一样。若目的操作数为无符号数，每右移一次，使目的操作数除以 2。但是，用这种方法做除法时，余数将被丢掉。
SAR 算术右移指令的功能与 SHR 很相似，移位次数也由计数值决定。每移位一次，目的操作数各位右移一位，最低位进人 CF，但最高位(即符号位)保持不变，而不是补 0。每移一次，相当于对带符号数进行除 2 操作。

18、REP 指令和 LOOP 循环指令有什么区别？

REP 指令属于重复前缀指令，一般与字符串传送指令连用，使得字符串指令重复执行，加快串运算指令的执行速度。CX 寄存器中存放待处理的字符串长度（字数或字节数），每重复执行一次，地址指针 SI 和 DI 都根据方向标志自动进行修改；CX 的值则自动减 1，直到整个字符串传送完毕，CX＝0 为止。而 LOOP 是循环控制指令，用于控制一个程序段的重复执行，指令执行前需将重复次数放入 CX 寄存器，每执行一次 LOOP 指令，CX 自动减 1，如果减 1 后 CX 不为 0,则转移到指令中所给定的标号处继续循环;直到 CX 为 0 时，则结束循环，转去执行 LOOP 指令之后的那条指令。所以这两条指令虽然都能实现重复执行，但重复执行的对象是不同的。

19、JMP 转移指令的类型和方式是什么？

JMP 指令是使程序无条件地转移到指令中指定的目的地址去执行。这类指令又分成两种类型：

第一种类型:段内转移或近(NEAR)转移，转移指令的目的地址和 JMP 指令在同一代码段中，转移时仅改变 IP 寄存器的内容，段地址 CS 的值不变。

第二种类型:段间转移，又被称之为远(FAR)转移，转移指令的目的地址和 JMP 指令不在同一段中，发生转移时， CS 和 IP 的值都要改变，也就是说，程序要转移到另一个代码段去执行。

不论是段内还是段间转移，就转移地址提供的方式而言，又可分为两种方式：

第一种方式:直接转移，在指令码中直接给出转移的目的地址，目的操作数用一个标号来表示，它又可分为段内直接转移和段间直接转移。

第二种方式:间接转移，目的地址包含在某个 16 位寄存器或存储单元中，CPU 必须根据寄存器或存储器寻址方式，间接地求出转移地址。同样，这种转移类型也可分为段内间接转移和段间间接转移两种方式。

所以 JMP 无条件转移指令可分成段内直接转移、段内间接转移、段间直接转移和段间间接转移这四种不同的类型和方式

20、RET 指令和 HIT 指令的作用分别是什么？

RET 指令是子程序（过程）的返回指令，它与 CALL 指令相呼应，使过程执行完毕后，能正确返回主程序中紧跟在 CALL 指令后面的那条指令继续运行，从而使过程调用返回。

RET 指令不带任何参数时，用于放在子程序的结束位置，被调用的子程序必须有 RET 指令，否则调用没有 RET 指令的子程序会导致自陷，子程序执行完之后会处于失控状态。

带参数 **RET n 指令**表示子程序返回主程序的同时，堆栈中再弹出 n 个字节的数据（栈顶指针加 n）。在 RET 指令上加弹出值 n 可以让调用过程的主程序通过堆栈向过程传递参数。这些参数必须在调用过程前推入堆栈，过程在运行中可以通过堆栈指针找到它们。当过程返回时，这些参数已没有用处，应该把它们从栈中弹出。使用带弹出值的 RET 指令后，可以在过程返回主程序时，使 SP 自动增量，从而不需要用 POP 指令就能把这些参数从堆栈中弹出。
而 **HIT 停机指令**是使 CPU 进入暂停状态，这时 CPU 不执行任何操作，直到系统复位或发生外部中断（NMI 引脚或 INTR 引脚上出现中断请求信号）为止，中断使 CPU 继续执行后面的指令。在程序中，通常用 HLT 指令来等待中断的出现。

21、过程调用的流程是什么？
过程调用有**近调用**和**远调用**两种类型。近过程调用指调用指令 CALL 和被调用的过程在同一代码段中，远过程调用则是指两者不在同一代码段中。CALL 指令执行时分两步进行：
第一步是将返回地址，也就是 CALL 指令下面那条指令的地址推入堆栈。对于近调用来说，执行的操作是：
SP<-SP-2，IP 入栈
对于远调用来说，执行的操作是：
SP<-SP-2，CS 入栈
SP<-SP-2，IP 入栈；
第二步是转到子程序的入口地址去执行相应的子程序。入口地址由 CALL 指令的目的操作数提供，寻找入口地址的方法与 JMP 指令的寻址方法基本上是一样的，也有四种方式:
段内直接调用，段内间接调用，段间直接调用和段间间接调用，但没有段内短调用指令。
执行过程中的 RET 指令后，从栈中弹出返回地址，使程序返回主程序继续执行。这里也分为两种情况：
如果从**近过程返回**，则从找中弹出一个字->IP，并且使 SP<-SP+2。
如果从**远过程返回**，则先从栈中弹出一个字->IP，并且使 SP<-SP+2;再从栈中弹出一个字->CS，并使 SP<-SP+2。

22、中断源的类型有哪些？
所谓**中断**是指计算机在执行正常的程序的过程中，由于某些事件发生，需要暂时中止当前程序的运行，转到中断服务程序去为临时发生的事件服务，中断服务程序执行完毕后，又返回正常程序继续运行，这个过程称为中断。

引起中断的原因称为中断源，8086 的中断源有两种：
第一种中断是**外部中断**，也称为硬件中断，它们从 8086 的不可屏蔽中断引脚 NMI 或可屏蔽中断引脚 INTR 引入。NMI 引脚上的中断请求，情况比较紧急，如存储器校验错、电源掉电等发生时， CPU 必须马上响应。从 INTR 脚上来的请求信号， CPU 可以立即响应，也可以暂时不响应。如果 CPU 内部标志寄存器中的 IF 置 1，则允许响应这类中断；若 IF 标志为 0，则不予响应。在 PC 机中，外部的可屏蔽中断是通过 8259A 可编程中断控制器连到 CPU 上的，主要用来管理 I/O 设备与 CPU 之间的通信。
第二种中断是**内部中断**，这种中断是为解决 CPU 在运行过程中发生一些意外情况，如除法运算出错和溢出错误等，或者是为便于对程序进行调试而设置的，如单步中断和断点中断。此外，也可以在程序中安排一条中断指令 INT n，利用这条指令直接产生 8086 的内部中断，指令中的 n 为中断类型号。

23、OFFSET 和 LEA 指令的区别？
OFFSET 操作符是在汇编阶段取得变量或标号所在段的段内偏移地址。**LEA 指令**在指令执行时计算出偏移地址。OFFSET 无需在执行时计算、指令执行速度更快。LEA 指令能获取汇编阶段无法确定的偏移地址。如 LEA BX，[BX+SI+0F62H]。且两者的源操作数类型不同。OFFSET 只能用于简单变量，即只能与简单的符号地址相连，而不能和诸如 6[SI]或[SI]等复杂操作数相连。如 MOV BX，OFFSET [SI+06H] 是错误的指令指针传送指令。

24、中断和一般的过程调用的概念和作用？
子程序（过程）是微机基本程序结构中的 1 种，基本程序结构包括顺序（简单）、分支（判断）、循环、子程序和查表等 5 种。
子程序是一组可以公用的指令序列，只要给出子程序的入口地址就能从主程序转入子程序。子程序在功能上具有相对的独立性，在执行主程序的过程中往往被多次调用，甚至被不同的程序所调用。一般微机首先执行主程序，碰到调用指令就转去执行子程序，子程序执行完后，返回指令就返回主程序断点（即调用指令的下一条指令），继续执行没有处理完的主程序，这一过程叫做主程序调用子程序过程。
子程序结构可简化程序，防止重复书写错误，并可节省内存空间。计算机中经常把常用的各种通用的程序段编成子程序，提供给用户使用。用户在编写程序时，只要会调用这些子程序，就可以大大简化用户编程的困难。
中断是计算机中央处理单元 CPU 与外设 I/O 交换数据的一种方式，

除此方式外，还有无条件、条件（查询）、存储器直接存取 DMA 和 I/O 通道等四种方式。由于无条件不可靠，条件效率低，DMA 和 I/O 通道两种方式硬件复杂，而中断方式 CPU 利用效率高，因此一般大多采用中断方式。

中断是指当计算机正在执行某一（主）程序时，收到一个中断请求。如果中断响应条件成立，计算机就把正在执行的程序暂停一下，去响应处理这一请求，执行中断服务程序。处理完服务程序后，中断返回指令 IRET 使计算机返回到原来还没有执行完的程序断点处继续执行，这一过程称为中断过程。有了中断，计算机才能具有并行处理，实时处理和故障处理等重要功能。

25、中断和一般的过程调用的联系和区别？

联系：中断与子程序调用两种过程属于完全不同的概念，但它们也有一些相似之处。首先，两者都需要保护断点（即下一条指令地址）、跳转至子程序或中断服务程序、保护现场、子程序或中断处理、恢复现场、恢复断点（即返回主程序）。其次，两者都可实现嵌套，即正在执行的子程序再调用另一子程序或正在处理的中断程序又被另一个新的中断请求所中断，嵌套可为多级。

区别：中断过程与子程序调用两种过程的相似点是表面的，从本质上讲两者是完全不一样的。

两者的根本区别主要表现在**服务时间**与**服务对象**上。

第一，子程序调用过程发生的时间是已知的和固定的，即在主程序中的调用指令 CALL 执行时，发生主程序调用子程序。调用指令所在的位置是已知和固定的。而中断过程发生的时间一般的随机的，CPU 在执行某一主程序时收到中断源提出的中断请求时，就发生中断过程，而中断申请一般由硬件电路产生，请求提出的时间是随机的（软中断发生时间是固定的），也可以说，子程序调用是程序设计者事先安排的，而执行中断服务程序是由系统工作环境随机决定的；

第二，子程序是完全为主程序服务的，两者属于主从关系，主程序需要子程序时就去调用子程序，并把调用结果带回主程序继续执行。而中断服务程序与主程序两者一般是无关的，不存在谁为谁服务的问题，两者是平行关系；

第三，主程序调用子程序过程完全属于软件处理过程，不需要专门的硬件电路。而中断处理系统是一个软、硬件结合系统，需要专门的硬件电路才能完全处理中断的过程；

第四，子程序嵌套可实现若干级，嵌套的最多级数由计算机内存开辟的堆栈大小所限制；而中断嵌套级数主要由中断优先级数来决定，一般优先级数不会很大。

26、汇编语言语句的种类有哪些？

汇编语言语句可以分为指令性语句、伪指令语句和宏指令语句。

指令性语句：与机器指令相对应，能产生目标代码，是 CPU 可以执行的、能完成特定功能的语句。

伪指令语句：没有对应的机器指令，汇编程序汇编源程序时对伪指令进行处理，不产生目标代码，仅在汇编过程中告诉汇编程序应如何完成数据定义、符号定义、段定义、段分配、存储器分配、过程定义、指示程序结束等操作。

宏指令语句：可把多次使用的同一个程序段定义为一条宏指令。一条宏指令包含

若干条指令。当一条宏指令作为语句出现时，该语句称为宏指令语句。调用时可

简单地用宏指令名来代替。在汇编时，遇到宏指令就用宏指令体中的指令来代替

这条宏指令语句。

27、EQU 伪指令和=伪指令的相似点和区别是什么？

赋值语句 EQU 格式：符号名 EQU 表达式

功能:用来给变量、标号、常数、指令、表达式等定义一个符号名，程序中用到 EQU 左边的变量、标号时可用右边的常数值或表达式值代替，但一经定义，在同一个程序模块中不能重新定义，EQU 指令不占内存单元。在 EQU 语句右边表达式中有变量或标号的表达式，必须先给他们定义，否则汇编程序将指示出错。

等号语句“=”与 EQU 语句具有相同功能，并且也不占内存单元。区别仅在于 EQU 中左边的标号不允许重新定义，而用“=”定义的语句允许重复定义，使符号具有新值。

28、伪指令助记符 DB、DW、DD 的区别是什么？

DB 定义字节，每个操作数占用一个字节空间。

DW 定义字，每个操作数占用一个字空间。存放时，低位字节存放在低地址单元，高位字节存放在高地址单元。

DD 定义双字，每个操作数占用两个字空间。存放时，低位字存放在低地址单元，高位字存放在高地址单元。

定义字符型数据时，用单引号“‘”括起来的一个或多个字符被称为字符串，用 DW 定义字符串时，只允许包括两个字符，多于两个字符时，只能用 DB 定义。用 DB 能定义任意多个字符，按书写顺序依次把 ASCII 码存入内存中。DW 只能定义一个或两个字符，按低地址在

低地址单元、高地址在高地址单元存放。注意伪指令 DB 和 DW 的区别，操作数都是‘AB’两个字符，对于用 DW 定义的字符串‘AB’，高字节字符是‘A’，所以它存在高地址，这与 DB 按字符顺序依次存入内存有所不同。

用伪指令 DW、DD 定义数据项时，操作数可以是变量名或标号名，DW 将其变量或标号的偏移地址存入存储区，而 DD 将其变量或标号的偏移地址和段地址存入存储区中，低位字(前两个单元)用于存放偏移地址，高位字(后两个单元)用于存放段地址。

29、常用的 DOS 键盘功能调用有哪些？

1)键入单字符

DOS 功能调用中 1，8，6，7 号功能调用都能完成从键盘输入一个字符到 AL 寄存器，差别在 1 号和 6 号功能调用键入同时在屏幕上显示字符，8 号和 7 号功能调用不回显。

(1) 1 号功能调用：从键盘输入字符并显示，调用命令为：

```
MOV   AH,1
INT   21H
```

执行上述命令后，系统扫描键盘等待有键按下，若有键按下，就将键值(ASCII 码)读入，先检查是否为 Ctrl-Break 键，若是就自动调用中断 INT 23H，执行退出命令，否则将键值送 AL 寄存器并在屏幕上显示此字符。

(2) 8 号功能调用：从键盘输入字符，但不回显，检查键入是否为 Ctrl-Break 键，调用命令为：

```
MOV   AH, 8
INT   21H
```

(3) 6 号功能调用：直接控制台输入/输出，6 号功能调用命令为：

```
MOV   DL, 0FFH
MOV   AH, 6
INT   21H
```

它可以从键盘输人字符，也可以向屏幕输出字符，并且不检查是否为 Ctrl-Break 键。当 DL=0FFH 时，表示从键盘输入，若标志位 ZF=0,AL 中为键值，若 ZF=1，表示无键按下，AL 中不是键值。DL 不等于 0FFH 时，表示屏幕输出。

(4) 7 号功能调用：直接控制台输入/输出，但无回显，不检查是否为 Ctrl-Break 键，命令格式为：

```
MOV AH, 7
INT   21H
```

2)输入字符串

0AH 功能调用：能从键盘接收字符串到内存的输人缓冲区，要求预先

定义一个输入缓冲区，缓冲区的第一个字节指出能容纳的最大字符个数，由用户给出;第二个字节存放实际输入的字符个数，由系统最后填人;从第三个字节开始存放从键盘接收的字符，直到 ENTER 键结束。

3)检验键盘状态

0BH 功能调用: 检验是否有键按下，若有键按下， AL= 0FFH，若没有键按下， AL=0，无论检测到是否有键按下，程序继续执行下一条指令。

4)清除键盘缓冲区

OCH 功能调用: 先清除键盘缓冲区，然后执行 AL 中指定的功能，AL 中可以指定 1,6,7,8 或 0AH 功能号，使程序在输入字符前将以前键入的字符清掉。

30、常用的 DOS 显示功能调用有哪些?

1)单字符显示

(1) 2 号功能调用: 2 号功能调用实现将字符送到屏幕显示出来。它要求将要显示字符的 ASCII 码值送人 DL 寄存器。

```
MOV   DL，‘：’
MOV   AH,   2
INT   21H
```

调用结果:屏幕上在光标位置处显示‘，。

(2) 6 号功能调用: 是直控制台输人/输出调用，除了前面谈到的键盘输入功能外，在 DL 不等于 0FFH 时，表示向屏幕输出，DL 中存放输出字符的 ASCII 码值。

```
MOV   DL, 24H      ‘$’的 ASCII 码值为 24H
MOV   AH，6
INT   21H
```

调用结果:屏幕在光标处显示‘$’字符。

2)字符串输出

9 号功能调用: 显示字符串，要求 **DS:DX** 指向串地址首址，并且字符串必须以‘$’字符为结束符。若要求显示字符串后光标自动回车换行，则在’$’字符前再加上 0DH(回车)，0AH(换行)字符。

31、宏指令与子程序的区别是什么?

宏汇编是用一条宏指令来代替一个程序段，可有效地缩短源程序的书写长度，且格式清晰，调用方便。在某种意义上，过程调用也有类似的功能，但二者之间有明显的区别，**主要区别**在以下几个方面：

(1) 过程调用使用CALL 语句，由CPU 执行，宏指令调用由宏汇编程序MASM 中宏处理程序来识别。

(2) 过程调用时，每调用一次都要保留程序的断点和保护现场，返

回时要恢复现场和恢复断点，增加了操作时间，执行速度慢。而宏指令调用时，不需要这些入栈及出栈操作，执行速度较快。
(3) 过程调用的子程序与主程序分开独立存在，经汇编后在存储器中只占有一个子程序段的空间，主程序转入此处运行，因此目标代码长度短，节省内存空间。而宏调用是在汇编过程中展开，宏调用多少次，就插入多少次，因此，目标代码长度长，占内存空间多。
(4) 一个子程序设计，一般完成某一个功能，多次调用完成相同操作，仅入口参数可以改变，而宏指令可以带哑元，调用时可以用实元取代，使不同的调用完成不同的操作，增加使用的灵活性。
如果多次调用的程序较长，速度要求不高，适宜用过程调用方法，如果多次调用的程序较短，而操作又希望修改，适宜用宏指令语句。

32、存储器的分类有哪些？
存储器可以分为内部存储器和外部存储器。
1）**内部存储器**也称为内存，是主存储器，位于计算机主机的内部，用来存放系统软件和当前正在使用的或者经常使用的程序和数据，CPU 可以直接对它们进行访问。内部存储器主要是半导体存储器，存取速度较快。内存的容量大小受到地址总线位数的限制，例如8086CPU 有20 条地址总线，可以寻址空间为1MB 。
内部存储器按照存储器类别又可以分为随机存取存储器 RAM 和只读存储器 ROM。
RAM 可以随机读写，断电后，所有存储数据全部消失。
按照集成电路内部结构的不同，RAM 又分为动态 RAM 和静态 RAM 两种。
ROM 只能读出已存储的内容，不能随意写入，已存储的内容由厂家或用户预先用设备写入，因此是非易失性的，断电后数据也不会消失。一般来说，ROM 中通常存储操作系统的程序或者用户固化的程序。
根据信息写入的方式不同，ROM 又可分成下面五种：
1. 掩膜型 ROM
ROM 中信息是在芯片制造时由厂家写人的，用户对这类芯片无法进行任何修改。
2. 可编程只读存储器 PROM
PROM 是厂家根据用户需求将芯片内二极管烧断而存储其内容，一般固化程序用，写入后不可更改。
3. 可擦除可编程只读存储器 EPROM
EPROM 用设备写人内容后，可由紫外光照擦除其内容，以便重新写入程序。

4. 电可擦除可编程只读存储器 EEPROM
EEPROM 用设备写入内容后，可由加电擦除其内容,芯片可反复使用。
5. 闪存(Flash Memory)
闪存用设备写入内容后，可由加电擦除其部分内容，芯片可反复使用。，闪存允许多线程重写，速度快，灵活性好。
2）**外部存储器**简称为外存，是辅助存储器。外存的特点是大容量、所存储的信息既可以修改，也可以保存，但是存取速度较慢，而且需要专门的设备来管理，比如，驱动器、控制芯片等。外存容量不受限制，也称为海量存储器。
外部存储器主要是磁记录存储器，典型的有软盘、硬盘。除此之外，还有光盘，磁带等。
3）**Flash 存储器**是最近兴起的一种新型存储器。Flash 存储器集成度高；是固态的存储介质，没有硬盘那样的机械装置，可靠性好，不易损坏；既有 SRAM 读写的灵活性和较快的访问速度，又有掉电后不丢失信息的特点。

33、静态 RAM 和动态 RAM 的特点和区别？
静态 RAM(SRAM)具有如下特点：
1）用双极型电路或 MOS 电路组成触发器作存储单元
2）由 6 个 MOS 管组成 1 位作为基本存储电路
3)难以构成大容量的存储器件，适合用于小容量存储器
4）存取速度较动态 RAM 快，存取时间可小到 2ns，CPU 的高速缓存(Cache)是由 SRAM 组成
5）访问速度快，访问周期达 20ns~40ns。
6）集成度高于双极型，但低于动态 RAM。
7）工作稳定，不需要刷新，故可省去刷新电路。
8）外部电路简单，但基本存储单元所包含的管子数目较多
9）功耗比双极型的低，但比动态 RAM 高。
10）只要有电源存在，其内容就不会消失，易于用电池作为后备电源。
动态 RAM(DRAM)具有如下特点：
1）密度高，但存取速度慢
2）它用 MOS 电路和电容作为存储单元，因此集成度很高
3）基本存储电路用单管线路组成(靠电容存储电荷来保存信息)
4）由于电容放电的效应，内容在一定时间后会自动消失，因此必须定时对其充电，称为刷新。
5）需要配置刷新逻辑电路，在刷新周期中，存储器不能执行读/写操作。
6）比静态 RAM 的功耗更低。

7）成本较低，价格比静态 RAM 便宜。
8）微机中的内存条是由 DRAM 组成
9）为了提高集成度，减少引脚的封装数， DRAM 的地址线分成行地址和列地址两部分。因此，在对存储器进行访问时，需要先后将行列地址送入行列地址锁存器。即地址需要两次打入。
10）在组成大容量存储器时作为主要使用器件。

34、Cache 的结构是什么？

Cache 一般由两部分组成，一部分存放由主存储器来的数据，另一部分存放该数据在主存储器中的地址(此部分称地址标记存储器，记为Tag)。由关联性，高速缓冲存储器结构可分为：全相联Cache；直接映象
Cache和成组相联Cache。

1）全相联Cache: 多数程序运行时需要访问位于主存储器不同位置的子程序，数据，堆栈及缓冲区等。因此，Cache中存储的块与块之间，存储块的地址之间或存储的顺序没有直接关系。全相联Cache不但保存许多互不相关的数据块，而且要同时存储每一个数据块
在主存储器中的地址（存放在Cache的地址标记存储器Tag中），当
CPU访问主存储器时，Cache控制器将访问主存储器的地址与Tag中存放的所有地址去逐一比较，匹配成功才命中。

2）直接映象Cache: 对直接映象Cache结构，地址仅需要比较一次就能确定是否命中。主要思想是将Cache中的每块分配一个索引字段，以确定块内寻址，用Tag字段区分存放在Cache中的主存储器位置。也就是说将主存储器分成若干页，主存储器中的每一页与Cache存储器大小相同，Cache中的Tag存储器保存主存储器的页号（页她址），
Cache中的索引字段保存主存储器在此页中的偏移地址，即主存储器的偏移量直接映象为Cache的偏移量。直接映象Cache在程序频繁访问不同页号主存储块时，Cache控制器要做频繁的转换，降低了系统性能，但由于它成本低，性能能够满足要求，亦被广泛使用。

3)成组相联Cache: 成组相联Cache结构结合了全相联Cache和直接映象Cache的特点，在Cache中具有多组直接映象的Cache，它们并行地起着高速缓存的作用。也就是说每组Cache中采用直接映象结构，组之间采用全相联结构。C

PU访问存储器时，Cache控制器要进行两次比较，才决定是否命中。目前计算机中常采用双路组相联或四路组相联高速缓冲存储器。

35、CPU与存储器连接时要考虑哪些问题？

在CPU对存储器进行读写操作时，首先在地址总线上给出地址信号，然后发出相应的读或写控制信号，最后才能在数据总线上进行数据交换，所以CPU与存储器的连接包括地址线、数据线和控制线的连接等三个部分。在连接时要考虑以下几个问题：

(1)CPU总线的负载能力

一般来说，CPU总线的直流负载能力可带一个TTL负载，目前存储器基本上是MOS电路，直流负载很小，主要负载是电容负载。因此在小型系统中，CPU可以直接和存储器芯片相连，在较大的系统中，考虑到CPU的驱动能力，必要时应加上数据缓冲器（例如74LS245)或总线驱动器来驱动存储器负载。

(2)CPU的时序和存储器存取速度之间的配合

CPU在取指令和读/写操作数时，有它自己固定的时序，应考虑选择何种存储器来与CPU时序相配合。若存储器芯片已经确定，应考虑如何实现Tw周期的插入。

(3) 存储器的地址分配和片选

内存分为ROM区和RAM区，RAM又分为系统区和用户区，每个芯片的片内地址，由CPU的低位地址来选择。一个存储器系统有多片芯片组成，片选信号由CPU的高位地址译码后取得。应考虑采用何种译码方式，实现存储器的芯片选择。

(4)控制信号的连接

8086CPU与存储器交换信息时，提供了以下几个控制信号：M/$\overline{IO}$，$\overline{RD}$，$\overline{WR}$，ALE，READY，$\overline{WAIT}$，DT/$\overline{R}$和$\overline{DEN}$，连接好这些信号与存储器要求的控制信号才能实现所需要的控制功能。

36、存储器的地址选择方式有哪些？

一个存储器系统通常由许多存储器芯片组成，对存储器的寻址必须有两个部分，通常是将低位地址线连到所有存储器芯片，实现片内寻址，将高位地址线通过译码器或线性组合后输出作为芯片的片选信号，实现片间寻址。由地址线的连接决定了存储器的地址分配。根据译码方式的不同，可以主要分为全译码法、部分译码法和线选法。

1. 全译码法

全译码法是将地址总线中除片内地址以外的全部高位地址接到译码器的输入端，以参与译码。采用该方法，每个存储单元的地址都是唯一的，不存在地址重叠，但译码电路复杂，连线也较多。

2. 部分译码法

部分译码法是将高位地址线中的一部分(而不是全部)进行译码，产生片选信号。该方法常用于不需要全部地址空间的寻址能力，但采用线选法地址线又不够用的情况。部分译码方式的可寻址空间比线性选择范围大，比全译码选择方式的地址空间要小。部分译码方式的译码器比较简单，但地址扩展受到一定的限制。

采用部分译码时，由于未参加译码的高位地址与存储器地址无关，所以存在地址重叠问题。当选用不同的高位地址线进行部分译码时，其译码对应的地址空间也不同。

3. 线选法

线选法是指高位地址线不经过译码，直接作为存储芯片的片选信号。每根高位地址线接一块芯片，用低位地址线实现片内寻址。该方法的优点是结构简单，缺点是地址空间浪费大,整个存储器地址空间不连续，而且由于部分地址线未参加译码，还会出现地址重叠。

总之，CPU与存储器相连时，将低位地址线连到存储器所有芯片的地址线上，实现片内选址。将高位地址线单独选址(线选法），或经过译码器(部分译码或全译码)译码输出控制芯片的选片端，以实现芯片间寻址。连接时要注意地址分布及重叠区。

37、计算机和外设之间信息交换的问题和解决方案？

1) 速度不匹配

CPU的速度很高，而外设的速度要低得多，而且不同的外设速度差异甚大，它们之中既有每秒钟能传送兆位数量级的硬磁盘，也有每秒钟只能打印百位字符的串行打印机或速度更慢的键盘。

2) 信号电平不匹配

CPU所使用的信号都是TTL电平，而外设大多是复杂的机电设备，往往不能用TTL电平所驱动，必须有自己的电源系统和信号电平。

3) 信号格式不匹配

CPU系统总线上传送的通常是8位、16位或32位的并行数据，而各种外设使用的信息格式各不相同。有些设备上用

的是模拟量，而有些是数字量或开关量;有些设备上的信息是电流量，而有些却是电压量;有些设备采用串行方式传送数据，而有些则用并行方式。

4) 时序不匹配

各种外设都有自己的定时和控制逻辑，与计算机的CPU时序不一致。因此，输人输出设备不能直接与CPU的系统总线相连，必须在CPU与外设之间设置专门的接口电路来解决这些问题。

接口电路是专门为解决CPU与外设之间的不匹配，不能协调工作而设置的，它处在总线和外设之间，一般应具有以下基本功能：

1) **设置数据缓冲以解决两者速度差异所带来的不协调问题**

CPU和外设间速度不协调的问题可以通过设置数据缓冲来解决，也就是事先把要传送的数据准备在那里，在需要的时刻完成传送。经常使用锁存器和缓冲器，并配以适当的联络信号来实现这种功能。

2) **设置信号电平转换电路**

外设和CPU之间信号电平的不一致问题，可通过在接口电路中设置电平转换电路来解决。

3) 设置信息转换逻辑以满足对各自格式的要求

由于外设传送的信息可以是模拟量，也可以是数字量或开关量，而计算机只能处理数字信号。因此需要进行数据格式的转换。这些工作均可由专门的接口电路来完成。

4) **设置时序控制电路来同步 CPU 和外设的工作**

接口电路接收 CPU 送来的命令或控制信号、定时信号，实施对外设的控制与管理，外设的工作状态和应答信号也通过接口及时返回给CPU，以握手联络信号来保证主机和外部 I/O 操作实现同步。

38、I/O 端口的种类有哪些？

CPU与外设通信时，传送的信息主要包括数据信息、状态信息和控制信息。在接口电路中，这些信息分别进入不同的寄存器，通常将这些寄存器和它们的控制逻辑统称为I/O端口，CPU可对端口中的信息直接进行读写。在一般的接口电路中都要设置以下几种端口：

1) 数据端口

数据端口用来存放外设送往CPU的数据以及CPU要输出到外设去的数

据。这些数据是主机和外设之间交换的最基本的信息，长度一般为1~2字节。数据端口主要起数据缓冲的作用。

2) 状态端口

状态端口主要用来指示外设的当前状态。每种状态用1位表示，每个外设可以有几个状态位，它们可由CPU读取，以测试或检查外设的状态，决定程序的流程。需要指出的是，除了状态端口中的内容外，接口电路中往往还会有若干状态线,它们用电平的高低来指示外设的当前状态，实现CPU与外设之间的通信联络，它们的名称和电平与状态位之间不一定完全对应。接口电路状态端口中最常用的状态位有：(1) 准备就绪位、(2) 忙碌位、(3) 错误位。

3) 命令端口

命令端口也称为控制端口，它用来存放CPU向接口发出的各种命令和控制字，以便控制接口或设备的动作。常见的命令信息位有启动位、停止位、允许中断位等。接口芯片不同，控制字的格式和内容是各不相同的，常见的控制字有方式选择控制字，操作命令字等。

39、I/O端口的寻址方式的特点？

CPU对外设的访问实质上是对I/O接口电路中相应的端口进行访问，因此和存储器那样，也需要由译码电路来形成I/O端口地址。I/O端口的编址方式有两种，分别称为存储器 映象寻址方式和I/O指令寻址方式。

1）存储器映像寻址方式

定义：该方式是指把I/O端口和存储单元统一编址，即把I/O端口看成是存储器的一部分,一个I/O端口的地址就是一个存储单元的地址。

优点：

① CPU访问存储单元的所有指令都可用于访问I/O端口， CPU访问存储单元的所有寻址方式也就是CPU访问I/O端口的寻址方式。

② 不用设置专门的I/O指令,微处理器的指令集中不必包含I/O操作指令，简化了指令系统的设计。

③ 能用类型多，功能强的访问存储器指令，对I/O设备进行方便、灵活的操作。

缺点：

①I/O端口占用了内存空间。

②是访问存储器还是访问I/O端口，在程序中不能一目了然。

2) I/O独立编址方式

定义：是指把I/O端口和存储单元各自编址，即使地址编号相同也无

妨。
优点：
① I/O 端口不占用内存空间；
② 访问 I/O 端口指令仅需两个字节，执行速度快；
③ 读程序时只要是 I/O 指令，即知是 CPU 访问 I/O 端口，使得程序清晰，可读性好。
④I/O 地址译码电路较简单。
缺点：
①要求 CPU 指令系统有独立的 I/O 指令，这些指令的功能没有访问存储器指令强；
②CPU 访问 I/O 端口的寻址方式少。
③CPU 还需提供能够区分访问内存和访问 I/O 的硬件引脚信号。

40、CPU 与外设之间的数据传送方式有哪些?

主机(CPU+内存)和外设之间数据传送的方式通常有 3 种：程序控制传送方式、中断传送方式和 DMA(直接存储器存取方式)。前两种方式主要由软件实现，DMA 方式主要由硬件实现。

1）程序控制传送方式

是由程序来控制 CPU 和外设之间的数据传送，可分为无条件传送和查询式传送。

①无条件传送方式(又称同步传送方式)

假设外设已作好传送数据的准备，因而 CPU 直接与外设传送数据而不必预先查询外设的状态。

适用场合：适用于外部控制过程的各种动作时间是固定的且是已知的场合。

优点：无条件传送是最简便的传送方式，它所需的硬件和软件都很少，且硬件接口电路简单。

缺点：这种传送方式必须在已知且确信外设已准备就绪的情况下才能使用，否则出错。

②查询式传送方式（条件传送方式）

进行数据传送前，程序首先检测外设状态端口的状态，只有在状态信息满足条件时，才能通过数据端口进行数据传送，否则程序只能循环等待或转入其它程序段。

适用场合：CPU 与外设工作不同步。

查询式传送方式的缺点：

CPU 要不断地查询外设，当外设没有准备好时，CPU 要等待，而许多外设的速度比 CPU 要慢得多，所以 CPU 的利用率不高。

2) 中断传送方式

通常是在程序中安排好在某一时刻启动外设，然后 CPU 继续执行其程序，当外设完成数据传送的准备后，向 CPU 发出中断请求信号，在 CPU 可以响应中断的条件下，CPU 暂停正在运行的程序，转去执行中断服务程序，在中断服务程序中完成一次 CPU 与外设之间的数据传送，传送完成后立即返回，继续执行原来的程序。

3) DMA 方式

是在外设和内存之间以及内存与内存之间开辟直接的数据通道，CPU 不干预传送过程，整个传送过程由硬件来完成，不需要软件介入。在 DMA 方式中，对数据传送过程进行控制的硬件称为 DMA 控制器。

41、可屏蔽中断处理过程是怎样进行的，不可屏蔽和软件中断呢？

可屏蔽中断处理的过程一般分成几步：中断请求;中断响应;保护现场;转入执行中断服务子程序;恢复现场和中断返回。

具体流程为：

首先，需要具备 CPU 响应中断的条件：

(1) 外设提出中断申请。

(2) 本中断位未被屏蔽。

(3) 本中断优先级最高。

(4) CPU 允许中断。

其次，当中断接口电路中的中断屏蔽触发器未被屏蔽时，外设可通过中断接口发出中断申请。外设发出中断请求的时间是随机的，而 CPU 在每条指令的最后一个机器周期的最后一个 T 状态去采样中断请求输入线 INTR，当 CPU 在 INTR 引脚上接收到一个有效的中断请求信号，而 CPU 内部中断允许触发器是开放的(开中断可用指令 STI 来实现)，则在当前指令执行完后 CPU 响应中断。

第三，CPU 响应中断后，对中断接口电路发出两个中断响应信号 INTA，中断接口电路收到第二个 INTA 以后，通过数据线向 CPU 送中断类型号。 CPU 再响应外部中断，并转入相应中断服务子程序的过程中，并自动依次做以下工作：

(1) 从数据总线上读取中断类型号，将其存入内部暂存器。

(2) 将标志寄存器 flags 的值入栈。

(3) 将 flags 中的中断允许标志 IF 和单步标志 TF 清 0,以屏蔽外部其它中断请求，及避免 CPU 以单步方式执行中断处理子程序。

(4)保护断点，将当前指令下面一条指令的段地址 CS 和指令指针 IP 的值入栈，使中断处理完毕后，能正确返回到主程序继续执行。

(5)根据中断类型号到中断向量表中找到中断向量，转人相应中断服务子程序。

(6)中断处理程序结束以后，从堆栈中依次弹出 IP、CS 和 flags，然后

返回主程序断点处，继续执行原来的程序。

然而在有些情况下，即使中断允许标志位 IF 为 1， CPU 也不能立即响应外部的可屏蔽中断请求，而是要再执行完下一条指令才响应外部中断。例如，发出中断请求时，CPU 正在执行封锁指令。如果执行向段寄存器传送数据的指令(MOV 和 POP 指令)，也要等下一条指令执行完后，才允许中断。当遇到等待指令或串操作指令时，允许在指令执行过程中进入中断，但在一个基本操作完成后响应中断。

对于**不可屏蔽中断请求**，不必判断 IF 是否为 1，也不是由外设接口给出中断类型号，从 NMI 引脚进入的中断请求规定为类型 2。在运行中断子程序过程中，若NMI引脚上有不可屏蔽中断请求进入，CPU仍能响应。

软件中断由程序设定，没有随机性，它不受中断允许标志位 IF 的影响，中断类型号由指令 INT n 中 n 决定。正在执行软件中断时，如果有不可屏蔽中断请求，就会在当前指令执行完后立即予以响应。如果有可屏蔽中断请求，并且 IF=1，也会在当前指令执行完后予以响应。

42、寻找中断源的方法有哪些？

寻找中断源可以用**查询中断**及**矢量中断**两种方法。

查询中断是采用软件查询方法，中断响应后启动中断查询程序，依次查询哪个设备的中断请求触发器为 1，检测到后，转向此设备预先设置的中断服务程序入口地址。此方法较简,单，但花费时间多，并且后面的设备服务机会少， 在 8086 系统中一般采用矢量中断方法。

矢量中断是将每个设备的中断服务程序的入口地址(矢量地址)集中，依次放在中断向量表中。当 CPU 响应中断后，控制逻辑根据外设提供的中断类型号查找中断向量表，然后将中断服务程序的入口地址送到段寄存器和指令指针寄存器， CPU 转入中断服务子程序。这样大大加快了中断处理的速度。

43、矢量中断和中断矢量分别是什么？

矢量中断 ：是根据 CPU 响应中断时取得中断处理子程序入口地址的方式而得名的，它是一种寻找中断源的方法。它提供一个矢量，指向中断处理子程序的起始地址。

中断矢量 ：就是中断处理子程序的起始地址。

中断矢量表：全部矢量放在内存的某一区域中，形成一个中断矢量表。

44、矢量中断中，中断类型号是怎么获取的？

(1) 对于除法出错，单步中断，不可屏蔽中断 NMI，断点中断和

溢出中断， CPU 分别自动提供中断类型号 0~4。

(2) 对于用户自己确定的软件中断 INT n，类型号由 n 决定。

(3) 对于外部可屏蔽中断 INTR，可用硬件电路(例如，通用并行接口芯片 8212)设计产生中断类型号。

(4) 对于外部可屏蔽中断 INTR，也可以用可编程中断控制器 8259A 获得中断类型号。

45、中断的优先级次序是什么？对于可屏蔽中断的优先级设定方法？

优先级从高到低的**次序**为：

内中断(除法错，INTO，INT n)

不可屏蔽中断(NMI)

可屏蔽中断(INTR)

单步中断

对**可屏蔽中断的优先级设定**有三种方法：

1.软件查询中断优先级

2.硬件查询优先方式——菊花链法

3.矢量中断优先级

46、8253 总结

1.方式概述：

方式 0 为计数结束产生中断；

方式 1 为可编程单次脉冲，输出负的单脉冲；

方式 2 为分频工作方式或比率发生器，输出连续的负脉冲序列；

方式 3 为方波发生器，输出连续的对称方波；

方式 4 位软件触发选通，方式 5 为硬件触发选通。输出一个负脉冲。

2.方式计数区别：

1）方式 0、1 和 4，计数初值装入计数器后，仅一次有效。而对于方式 2、3 和 5，在减 1 计数到 0 值后，8253 会自动将计数值重新装入计数器。

2）除了方式 1、5 需外部触发才开始计数，其余方式在写入计数值后，自动开始计数。

3）方式 2 和方式 3 都能连续计数，可以自动循环计数。

3.方式触发区别：

1)软件触发只有方式 0 和方式 4，硬件触发只有方式 1 和方式 5，既可以采用软件也可以采用硬件触发的是方式 2 和方式 3.

2）门控信号 GATE（边沿）触发为硬件触发，电平触发为软件触发。

4.方式 2 和 3 总结：

1）方式 2 一般产生连续的负脉冲序列信号，且波形是不对称的。方式 3 产生的是对称的方波或者基本对称的矩形波。

2）产生周期性的中断信号，可选择方式 2 或者方式 3.

5.计数值格式区别：

1）计数值可以是二进制数或者十进制数（BCD 码）。

2)二进制数的取值范围是 0000H~FFFFH，表示的最大值是 65536。

3）十进制数的取值范围是 0000~9999，表示的最大值是 9999.

4）两者都是写入计数值 0000H 时，代表计数的最大值、定时时间最长。

6.初始化编程注意：

1）初始化编程时，先向控制寄存器中写入通道控制字，再向计数寄存器中写入计数初值。

2）每个计数器都有一个控制寄存器，用来保存该计数器的控制信息。控制寄存器只能写入，不能读出。

3）CPU 用输出指令向计数寄存器写入初值，用输入指令读回输出锁存寄存器中的数值。

47、8255 总结：

1.方式概述：

方式 0：基本输入输出方式，可实现无条件传送和查询传送数据，还可对 C 端口实现按位操作。

方式 1：选通或应答输入输出方式，可用于中断方式传送和查询传送数据。

方式 2：双向输入输出方式，只有 A 端口具有这种方式，方式 2 的功能相当于方式 1 的输入输出功能的结合。A 端口工作于方式 2 时，B 端口可按方式 0 或方式 1 工作。

2.控制字介绍：

1）8255 有工作方式选择控制字和 C 口按位置位/复位控制字两种控制字。

2）C 口按位置位/复位控制字必须跟在工作方式选择控制字之后写入控制字寄存器，即使仅使用该功能，也应先选送一个方式控制字。

3）方式选择控制字只需写入一次，而 C 口按位置位/复位控制字可多次使用。

4）设置方式控制字时，A 端口、B 端口作为整体设置，而 C 端口要分成上下两部分分别设置，但 3 个端口的工作方式均由一个控制字规定。

3.端口方式总结：

1）端口 A 有方式 0、1、2 三种工作方式，端口 B 只能工作于方式 0 和 1，而端口 C 仅工作于方式 0.
2）端口是输入方式时，对应 IN 指令，数据从端口—>CPU，相当于 RD，读出。
3）端口是输出方式时，对应 OUT 指令，数据从 CPU—>端口，相当于 WR，写入。

48、串行通信和并行通信的区别？

计算机与外部的信息交换称为通信，基本的通信方式有两种，一种是**并行通信**，另一种是**串行通信**。
1）**并行通信**时，数据各位同时传送。这种方式传输数据的速度快，但使用的通信线多，如果要并行传送 8 位数据，需要用 8 根数据线，另外还要加上一些控制信号线。随着传输距离的增加，通信线成本的增加将成为突出的问题，而且传输的可靠性随着距离的增加而下降。因此并行通信适用于近距离传送数据的场合。
2）在远距离通信时，一般都采用**串行通信**方式。它具有需要的通信线少和传送距离远等优点。
串行通信时，要传送的数据或信息必须按一定的格式编码，然后在单根线上，按一位接一位的先后顺序进行传送。发送完一个字符后，再发送第二个。接收数据时，每次从单根线上一位接一位地接收信息，再把它们拼成一个字符，送给 CPU 作进一步处理。当微机与远程终端或远距离的中央处理机交换数据时，都采用串行通信方式。
采用串行通信的另一个出发点是，有些外设，如调制解调器(MODEM)、鼠标器等，本身就需要用串行方式通信，因为这些设备是以串行方式存取数据的。此外，有些外设，如打印机、绘图仪等，除了使用并行方式外，也可采用串行方式与计算机通信。

49、串行通信中的数据传送方向？

串行通信时，数据在两个站(或设备)A 与 B 之间传送，按传送方向可分成单工、半双工和全双工三种不同的方式。

1. 单工

单工数据线仅能在一个方向上传输数据，两个站之间进行通信时，一边只能发送数据，另一边只能接收数据。现在把这种通信方式称为单向通信。

2. 半双工

在半双工方式中，数据可在两个设备之间向任一个方向传输，但两个设备之间只有一根传输线，故同一时间内只能在一个方向上传输数据，不能同时收发数据，如无线电对讲机。

3. 全双工

如果在一个数据通信系统中，对数据的两个传输方向采用不同的通路，这样的系统就可以工作在全双工方式。采用全双工的系统可以同时发送和接收数据，电话系统就是全双工传送数据的一个例子。计算机的主机和显示终端(它由带键盘的 CRT 显示器构成)进行通信时，通常也采用全双工方式。

50、数字调制的主要形式有哪些？

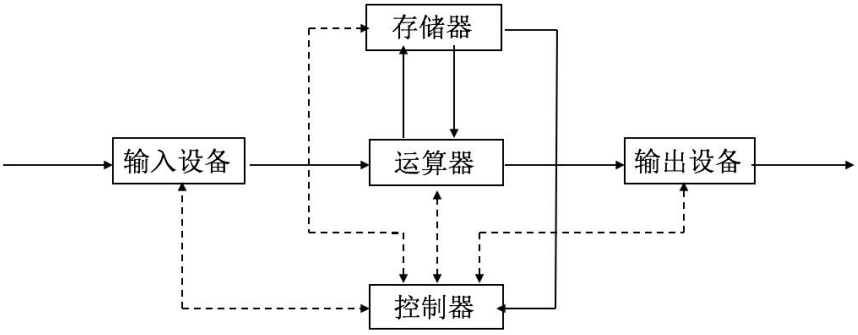
主要有幅度调制或幅移键控 **ASK** 简称“调幅”，频率键移 **FSK** 简称“调频”，相位键移 **PSK** 简称“调相”和**多路载波**等几种。

幅度调制用改变信号幅度的方法来表示数字信号 0 和 1。

频率键移调制用一种频率信号表示数字 0，另一种频率信号表示数字 1。

3.3.3 计算机组成原理

1、典型的冯.诺依曼计算机是以运算器为中心的，结构如图所示：

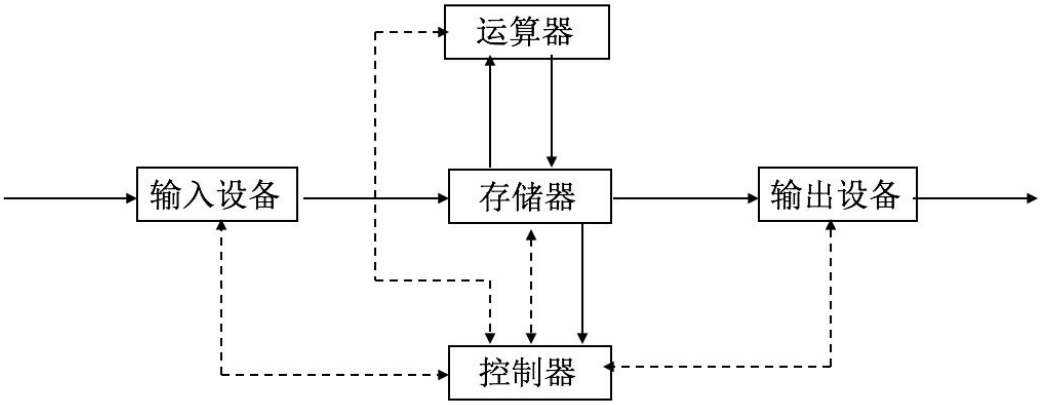


其中，实线是数据流，虚线为控制流

它的特点可归结如下：

- 计算机由五个部件构成：运算器、控制器、存储器、输入设备、输出设备
- 指令和数据“以同等地位”存放于存储器内，分别按地址访问
- 指令和数据均用二进制码表示
- 指令由操作码和地址码/操作数构成，操作码用来表示操作的性质，地址码用来表示操作数在存储器中的位置
- 指令在存储器内按顺序存放，通常，指令是顺序执行的，在特定条件下，可根据运算结果或根据设定的条件改变执行顺序。
- 机器以运算器为中心，输入输出设备与存储器间的数据传送通过运算器完成

2、现代的计算机以存储器为中心，结构如图所示：



其中，实线是数据流，虚线为控制流

各部件的功能如下：

- 运算器用来完成算术运算和逻辑运算，并将运算的中间结果暂存在运算器内
- 存储器用来存放数据和程序
- 控制器用来控制、指挥程序和数据的输入、运行以及处理运算结果
- 输入设备用来将人们熟悉的信息形式转换为机器能识别的信息形式，常见的有键盘、鼠标等
- 输出设备可将机器运算结果转换为人们熟悉的信息形式，如打印机输出、显示器输出等

3、计算机的工作过程是什么？

执行每一条指令，都包括取指、译码和执行三个基本步骤，所以，计算机的工作过程，也就是不断地取指令、译码和执行的过程，直到遇到停机指令。

具体如下：

取指：根据 PC（程序计数器）访存读取当前要执行的指令

PC＋1

译码：识别指令字中的操作类型，产生相应的控制信号

取操作数：根据指令字的地址域访存

执行

写回

4、程序设计语言的种类有哪些？

- 机器语言

  —由 0/1 代码表示机器能完成的各种操作（指令）

  —依赖于机器（处理器指令集），难于理解

- 汇编语言

  —用便于书写和记忆的符号表示机器的各种操作

—依赖于机器

- 高级语言

—描述方式适合人类的习惯

—与平台无关性

机器语言和汇编语言程序由各种操作、控制序列构成；

高级语言程序是面向问题描述的

5、计算机的处理过程是什么？

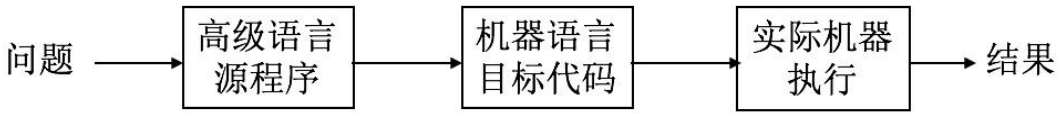
用户采用程序设计语言描述问题的求解过程；

计算机在程序的控制下完成问题的求解；

—计算机只能识别用 0/1 代码表示的程序

—用户需要使用高级语言编程

处理过程如下：



因此，需要将高级语言程序转换成机器语言程序，才能在实际机器上执行

—编译执行：利用编译器一次性将高级语言程序翻译成机器语言程序

如：C、FORTRAN

—解释执行：语句翻译一条，执行一条，直至结束

如：BASIC、JAVA

6、二进制编码器与译码器？

- 编码：是指用文字、符号和数码来表示某种信息的过程。
- 编码器：实现编码的数字电路，把输入的每个高低电平状态信号编成一组对应的二进制代码。
- 二进制编码器：对二进制编码的组合电路。

假设某编码器有 n 个输入端,$I_0, I_1 \ldots, I_{n-1}$,有 m 个输出端 $Y_0, Y_1 \ldots, Y_{m-1}$ 。为了不使输出发生混乱而产生错误，普通编码器规定，在任何给定的时刻，n 个输入端中只能有一个有效电平，其余 n-1 个都不能出现。同时也可以知道，输入端的个数与输出端的个数有以下关系：$2^m>=n$

- 译码：把代码状态的特定含义翻译过来的过程为译码。
- 译码器：实现译码操作的逻辑电路。
- 一般说来，输出编码比输入编码位数多。

设二进制译码器的输入端为 n 个，则输出端为 2^n 个，且对应于输入代码的每一种状态，2^n 个输出中只有一个为 1（或为 0），其余全

为 0（或为 1）。二进制译码器可以译出输入变量的全部状态，故也称为变量译码器。

7、数值的原码、反码、补码及它们的关系？

原码性质：

原码为符号位加数的绝对值，0 正 1 负

原码零有两个编码，+0 和-0 编码不同

原码难以用于加减运算

n+1 位二进制原码所表示的范围：

小数：MAX=$1-2^{-n}$ ，MIN= $-(1-2^{-n})$

整数：MAX=2^n-1 ，MIN= $-(2^n-1)$

反码性质：

反码零有两个编码，+0 和-0 编码不同

n+1 位二进制反码所表示的范围：

小数：MAX=$1-2^{-n}$ ，MIN= $-(1-2^{-n})$

整数：MAX=2^n-1 ，MIN= $-(2^n-1)$

补码性质：

0 的补码是唯一的

补码便于加减运算：

n+1 位补码所能表示的范围：

小数：MAX=$1-2^{-n}$ ，MIN= -1

整数：MAX= 2^n-1 ，MIN= -2^n

三者关系：

1）正数的 原码，反码，补码表示均相同，符号位为 0，数值位等于数的真值。

2）零的原码和反码均有 2 个编码，补码只一个码

3）负数的 原码，反码，补码表示均不同，符号位为 1，数值位：原码为数的绝对值；

4）负数的反码为原码每一位均取反；负数的补码为其反码再在最低位+1；

8、为什么补码便于加减运算？

有了补码表示的被加（减）数和加（减）数,要完成计算补码表示的二数之和或二数之差,只需用二数的补码直接执行加减运算即可,符号位与数值位同等对待,一起参加运算,若运算结果不溢出,即不超出计算机所能表示的范围,则结果的符号位和数值位同时为正确值.此外,还可以看到,实现减运算时,用的仍是加法器线路,把减数的负数的补

码送加法器即可。

9、运算结果的溢出判断方法？

- 如果运算的结果，超出了计算机能表示的数的范围，会得出错误的结果，这种情况称为溢出。
- 对于字长为 n 的计算机，那么它能表示的定点补码范围为$-2^{n-1} \leq X \leq 2^{n-1}-1$，若运算结果小于$-2^{n-1}$或大于$2^{n-1}-1$，则发生溢出，发生溢出时数值的有效位占据了符号位。

两种方法：

用一位符号位判断溢出

用两位符号位判断溢出

一位符号位：

通常用符号位产生的进位和最高有效位向符号位产生的进位进行异或操作后，按其结果进行判断。

—若异或结果为 1，则溢出；

—若异或结果为 0，则没有溢出。

两位符号位：

- 变形补码：

$$[x]_{补'}=\begin{cases} x & 1>x\geq 0 \\ 4+x & 0>x\geq -1 (\mathrm{mod}\ 4) \end{cases}$$

- 用变形补码做加法操作时，两位符号位连同数值部分一起参加运算。
- 溢出判断规则：正常时两个符号位的值相同，在运算结果中当两个符号位不同时则表明发生了溢出。

双符号含义：

00 表示运算结果为正数；

01 表示运算结果正溢出；

10 表示运算结果负溢出；

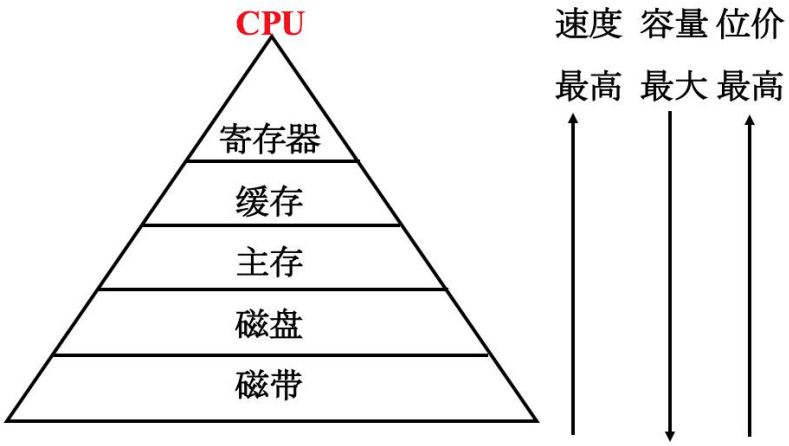
11 表示运算结果为负数。

第一位符号位为运算结果的真正符号位。符号位的低位是数值位的进位。

10、存储器的层次结构

存储器有三个主要特性：速度、容量、位价（即“价格/位”）。

层次结构如图所示：



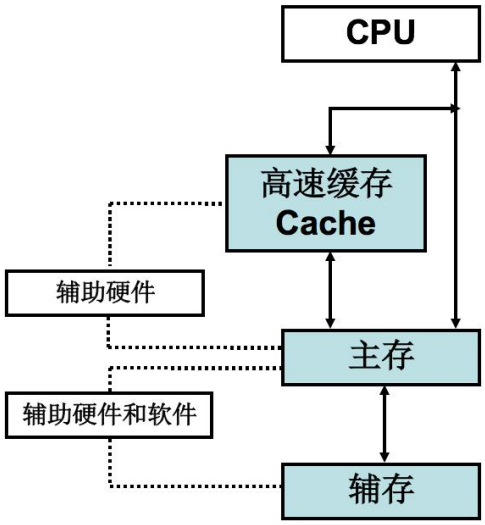
11、 CPU 的三级存储体系结构

三级存储系统：缓存—主存—辅存

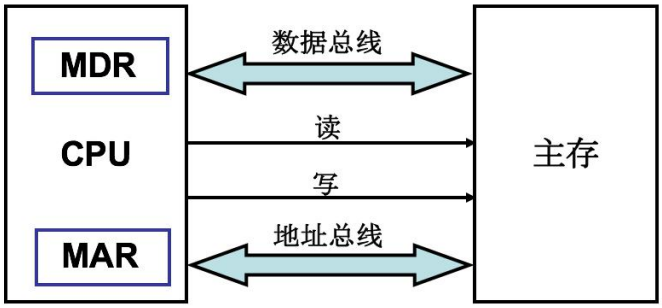
主存－缓存层次：弥补主存速度的不足，主要由专用硬件实现。

主存－辅存层次：弥补主存容量的不足，由硬件和软件实现。

结构如下：



12、主存和 CPU 的联系



存储器芯片封装了驱动器、译码器、读写电路等。而 MAR 和 MDR 则制作在 CPU 芯片中。

计算机中的主存数据寄存器(MDR)，与主存地址寄存器(MAR)帮助完

成 CPU 和主存储器之间的通信，MAR 用来保存数据被传输到的位置的地址或者数据来源位置的地址。MDR 保存要被写入地址单元或者从地址单元读出的数据。

13、存储容量的扩展

存储容量的扩展： 当一片 RAM 存储芯片不能满足存储容量需要时，就需要将若干片 RAM 存储芯片组合起来，构成满足存储容量要求的存储器。

三种扩展方法：

1.位扩展：增加存储字长

2.字扩展：增加存储字的数量

3.字位扩展：同时增加存储字长和存储字的数量

位扩展法：

- 仅在字长（位数）扩展，字数不做扩展。
- 字数满足要求，而位数不够时，应采用位扩展。

字扩展法：

- 只扩充字的数量，而位数不变。
- 在 RAM 的数据位的位数足够，而字数达不到要求时，需要进行字扩展。字数增加，地址线数就得相应增加。如 256×8 位 RAM 的地址线数为 8 条，而 1024×8 位 RAM 的地址线数为 10 条

字位扩展法:

假定存储器的容量为 M*N 位，若使用 L*K 芯片（L<M,K<N），共需要（M/L）*(N/K)个存储芯片，需要在字和位上同时扩展。

14、存储器与 CPU 的连接

1. 地址线的连接

低位地址线与存储芯片连接；高位地址线或用作存储芯片扩充时用，或作其他用法，如片选信号等。

2. 数据线的连接

必要时需要对存储芯片进行位扩展，使其数据位与 CPU 的数据线相等。

3. 读/写命令线的连接

直接与存储芯片的读写控制端相连，通常是高电平为读，低电平为写。

4. 片选线的连接

片选信号的连接是 CPU 与存储芯片正确工作的关键。

片选有效信号与 CPU 的访存控制信号 MREQ（低电平有效）有关。

MREQ 为低，表示 CPU 访问存储器；MREQ 为高，表示 CPU 访问 I/O，此时不要求存储器工作。

片选信号与地址的高位有关，未与存储芯片地址线连上的高位地址与访存控制信号共同作用产生存储器的片选信号。

5. 合理选择存储芯

类型 ROM 或 RAM、数量的选择。

15、提高访存速度措施

- 存储器带宽需求

对于 200MIPS 系统，总存储器带宽需要 505MW/s，要求访存周期 19.8ns，而一般的 DRAM 访存周期约 200ns

取指：200MW/s（200MIPS）

取数和写回：300MW/s(两个 OPR，一个结果),OPR 指操作数

I/O：5MW/s, (MW/s：兆字，设一条指令一个字)

- 采用高速主存
- 采用双端口存储器
- 并行访问存储器，提高存储器字长

—单体多字系统

—多体并行系统（交叉存储器）

- 采用层次化存储系统结构

16、单体多字系统和多体并行系统是什么？

单体多字系统：

一个访存周期内，从同一地址读出 4 条指令，然后再逐条将指令送至 CPU 执行。

要求指令和数据在主存中是连续存放的，否则产生“访存冲突”

多体并行系统：

大容量的主存，可由多个存储体组成，每个存储体都具有自己的读写线路、地址寄存器和数据寄存器。它们能并行工作，也能交叉工作。

- 并行访问

—同时访问 M 个模块，同时启动，同时读出，完全并行工作（不过，同时读出的 M 个字在总线上需分时传送）。

- 交叉访问

—M 个模块按一定的顺序轮流启动各自的访问周期

—启动两个相邻模块的**最小时间间隔**等于单模块访问周期 1/M

- 两种方式

—高位交叉编址

—低位交叉编址

17、Cache 的命中率和效率？

Cache 的出现主要是使 CPU 可以不用直接访问主存，只与高速 Cache 交换信息。

Cache 命中

—CPU 欲访问的数据已在缓存中，即可直接访问 Cache

Cache 不命中

—CPU 欲访问的数据不在 Cache 内，此时需将该数所在的主存整个子块一次调入 Cache 中。

命中率是指 CPU 要访问的信息已在 Cache 内的比率。通常用命中率来衡量 Cache 的**效率**。

Cache 效率

- Cache 的**容量**和**块长**是影响 Cache 效率的重要因素。
- Cache 容量越大，命中率越高。

—当 Cahce 容量达到一定值时，命中率不会因容量的增大而明显提高。

—Cache 容量大，成本增加。

—Cache 容量是总成本价与命中率的折衷值。

18、单一缓存和二级缓存

单一缓存：在 CPU 和主存之间只设一个缓存。

—片内缓存（片载缓存）：让出存储总线

—速度快、容量受限。

两级缓存：在主存和片内缓存之间在加一级缓存（即片外缓存）。

—这种由片外缓存和片内缓存组成的 Cache，叫做两级缓存，并称片内缓存为第一级，片外缓存为第二级。

19、统一缓存和分开缓存

统一缓存：指令和数据存放在同一个 Cache 内

分开缓存（Harvard 结构）：指令和数据分别存放在 I_Cache(指令 Cache)和 D_Cache(数据 Cache)内。

选用时要考虑的两个主要因素：

1. 与**主存结构**有关：如果计算机的主存是统一的（指令和数据在同一主存内），则相应的 Cache 采用统一缓存；如果主存采用指令、数据分开存放的方案，则相应的 Cache 采用分开缓存。
2. 与机器对指令执行的**控制方式**有关：当采用**超前控制**或**流水线控制**方式时，一般都采用分开缓存。

20、超前控制和流水线控制是什么？

- **超前控制：**是指在当前指令执行过程尚未结束时，就提前将下一条准备执行的指令取出，这一过程称为超前取指或者叫作指令预取。
- **流水线控制：**实质上是多条指令同时执行。
- 超前控制和流水线控制特别强调指令的预取和指令的并行执行。因此，这类机器必须将指令 Cache 和数据 Cache 分开，否则可能出现取指和执行过程对**统一缓存**的争用。

如果采用**统一缓存**，执行部件向缓存发出取数请求时，一旦指令预取机构也向缓存发出取指请求，那么统一缓存只有先满足执行部件要求，将数据送到执行部件，而取指请求暂时等待，显然达不到预取指令的目的。

21、Cache—主存地址映像方式

由主存地址映象到 Cache 地址称为**地址映象**。有四种实现方式：

直接映象：

- **全相联映象**
- **组相联影响**
- **段相连映象**

22、数据存储的顺序？

大端模式：是指数据的高字节保存在内存的低地址中，而数据的低字节保存在内存的高地址中，这样的存储模式类似于把数据当作字符串顺序处理：地址由小向大增加，而数据从高位往低位放；这和我们的阅读习惯一致。

小端模式：是指数据的高字节保存在内存的高地址中，而数据的低字节保存在内存的低地址中，这种存储模式将地址的高低和数据位权有效地结合起来，高地址部分权值高，低地址部分权值低。

目前 Intel 的 80x86 系列芯片是唯一还在坚持使用小端的芯片，ARM 芯片默认采用小端，但可以切换为大端；而 MIPS 等芯片要么采用全部大端的方式储存，要么提供选项支持大端—可以在大小端之间切换。另外，对于大小端模式的处理也和编译器的实现有关，在 C 语言中，默认是小端（但在一些对于单片机的实现中却是基于大端），Java 是平台无关的，默认是大端。在网络上传输数据普遍采用的都是大端。

23、指令和数据的寻址方式？

块寻址：

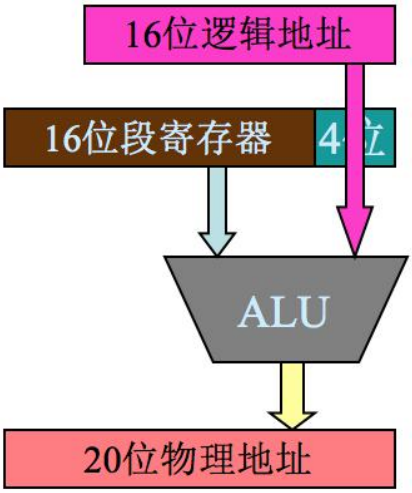
- 通常在指令中指出数据块的起始地址和数据块的长度，常用在输入输出指令中。

- 多用于 I/O 指令。对顺序连续的成块数据字进行寻址。
- 目的：压缩程序的长度，加块执行速度。
- 用于：1）两个部件间的数据交换；
  2）程序，数据块的浮动。

若块的长度可变，格式如下：

段寻址：

以 8086 的段寻址为例。这种寻址方式的实质是基值寻址。Intel 8086/8088 微机中，ALU 是 16 位运算，但其寻址范围可达到 1M，即地址为 20 位。



物理地址为 20 位，逻辑地址为 16 位
物理地址=段地址*10H+偏移地址

24、保护模式和实模式？

保护模式:寻址采用32位段地址和偏移量，最大寻址空间4GB，最大分段4GB (Pentium Pre 及以后为 64GB)。 在保护模式下CPU可以进入虚拟8086方式，这是在保护模式下的实模式程序运行环境。

保护模式下程序的运行: 保护模式—从程序运行说起无论实模式还是保护模式，根本的问题还是程序如何在其中运行。

因此我们在学习保护模式时应该时刻围绕这个问题来思考。和实模式下一样，保护模式下程序运行的实质仍是“CPU 执行指令，操作相关数据”，因此实模式下的各种代码段、数据段、堆栈段、中断服务程序仍然存在，且功能、作用不变。那么保护模式下最大的变化是什么呢?答案可能因人而异，我的答案是“地址转换方式”变化最大。

实模式:实模式是指寻址采用和8086相同的16位段地址和偏移量，最大寻址空间 1MB，最大分段 64KB。可以使用 32位指令。32位的x86CPU用作高速的 8086。

实模式下程序的运行 ：程序运行的实质是什么?其实很简单,就是指令的执行,显然CPU是指令得以执行的硬件保障,那么CPU如何知道指令在什么地方呢?

对了，80x86系列是使用CS寄存器配合IP寄存器来通知CPU指令在内存中的位置.程序指令在执行过程中一般还需要有各种数据,80x86系列有 DS、ES、FS、GS、SS 等用于指示不同用途的数据段在内存中的位置。

程序可能需要调用系统的服务子程序，80x86系列使用中断机制来实现系统服务。总的来说，这些就是实模式下一个程序运行所需的主要内容(其它如跳转、返回、端口操作等相对来说比较次要。)

两者区别:
保护模式和实模式的**根本区别**是进程内存受保护与否。可寻址空间的区别只是这一原因的结果。
实模式将整个物理内存看成分段的区域,程序代码和数据位于不同区域，系统程序和用户程序没有被区别对待，而且每一个指针都是指向"实在"的物理地址。这样一来，用户程序的一个指针如果指向了系统程序区域或其他用户程序区域，并且改变了值，那么对于这个被修改的系统程序或用户程序，其后果很可能是灾难性的。
为了克服这种低劣的内存管理方式，处理器厂商开发出**保护模式**。这样一来，物理内存地址不能直接被程序访问，程序内部的地址（虚拟地址）要由操作系统转化为物理地址去访问，程序对此一无所知。至此，进程（这时我们可以称程序为进程了）有了严格的边界，任何其他进程根本无法访问不属于自己的物理内存区域；甚至在自己的虚拟地址范围内也不是可以任意访问的，因为有一些虚拟区域已经被放进一些公共系统运行库，这些区域也不能随意修改。
CPU 启动环境为 16 位实模式，之后可以切换到保护模式。但从保护模式无法切换回到实模式。

25、CISC 和 RISC
CISC：复杂指令系统
- CISC 的问题

—庞大的指令集

—纷繁复杂的寻址模式
—硬件实现复杂（硬件资源的利用率低）

RISC：精简指令系统

- 使用 RISC 的目的—减小代码空间
- 精简指令集结构的特征

—每周期一条指令
—寄存器-寄存器操作（除 Load/Store 类型结构）
—简单的寻址方式
—简单的指令格式

- RISC 指令系统的最大特点

⑴选取使用频率最高的一些简单指令，指令条数少；
⑵指令长度固定，指令格式种类少；
⑶只有取数 / 存数指令访问存储器，其余指令的操作都在寄存器之间进行。

26、指令的执行过程

1）读取指令
—指令地址送入主存地址寄存器
—读主存，读出内容送入指定的寄存器
2）分析指令
3）按指令规定内容执行指令
—不同指令的操作步骤数
—和具体的操作内容差异很大
4）检查有无中断请求
若无，则转入下一条指令的执行过程

27、指令周期的特点是什么？

定义：执行一条指令所需的时间

- 通常，不同指令所需时间不同（如乘法操作的执行周期较长）
- **单周期实现**

—定长单周期：一个周期一条指令，周期宽度以数据通路最长的指令为准
—不定长单周期：一个周期一条指令，对于各个指令，周期宽度不同

- **多周期实现**

—将指令的执行过程划分成多个步骤，每个周期完成一个步骤。n 个 CPU 工作周期构成一个指令周期。

- 如取指周期、间址周期、执行周期、中断周期等

CPU 中设置标志触发器指示当前的工作周期

28、指令周期 vs 机器周期 vs CPU 工作周期 vs 时钟周期 vs 总线周期

机器周期：也称为 CPU 工作周期。通常用内存中读取一个指令字的最短时间来规定 CPU 周期。指令周期常常用若干个 CPU 周期数来表示；如取指、间址、执行周期。

时钟周期（T1/T2/T3/T4/Tw）：一个 CPU 周期时间又包含有若干个时钟周期，通常称为节拍脉冲或 T 周期，它是处理操作的最基本单位。

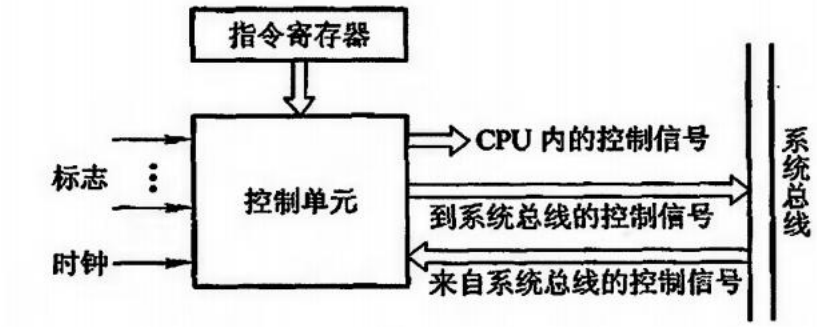
指令周期：CPU 每取出并执行一条指令，都要完成一系列的操作，这一系列操作所需要用的时间就是指令周期。

总线周期：8086 CPU 中，BIU 完成一次访问存储器或 I/O 端口操作所需要的时间。每个总线周期至少包含 4 个时钟周期（T_1~T_4），一般情况下，在总线周期的 T_1 状态传送地址，T_2~T_4 状态传送数据。

29、控制单元(CU)的功能？

功能：在时钟控制下，产生完成各个微操作所需的控制信号

微操作：各个机器周期所包含的动作



控制单元外特性

CU 的外特性：

输入信号

- 时钟
- 指令寄存器（IR）的 OP（操作码）字段
- 标志：指示 CPU 的当前状态
- 来自系统总线（控制总线）的控制信号，如：中断请求、DMA 请求。

输出信号

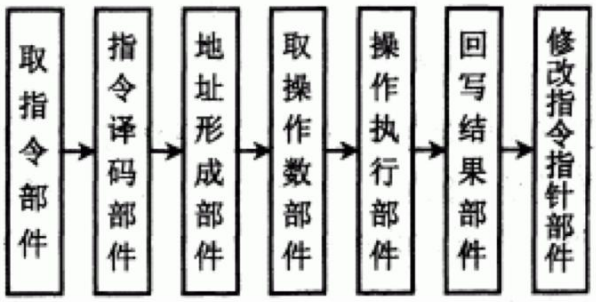
- 微操作控制信号
- 总线控制信号（访存、I/O 读/写、中断响应等）

30、流水线技术是什么？

流水过程由多个相互联系的**子过程**组成，每个子过程称为流水线的“级”或“段”。

—1 条流水线的段数被称为流水线的深度，或“流水深度”。

各个功能段所需时间尽量相等，否则，消耗时间长的功能段将成为流水线的瓶颈，会造成流水线的阻塞或断流。
—每个功能段所需的时间一般为一个机器周期。
指令流水线是将指令的整个执行过程用流水线进行分段处理。
指令流水线对机器性能的改善程度取决于把处理过程分解成多少个相等的时间段数。
指令流水线结构如下：



它将指令的处理过程分解为更细的几个阶段。

- **取指（FI）：**从存储器取出一条指令并暂时存入指令部件的缓冲区。
- **指令译码（DI）：**确定操作性质和操作数地址的形成方式。
- **计算操作数地址（CO）：**计算操作数的有效地址，涉及到寄存器间址、间址、变址、基址、相对寻址等各种地址计算方式。
- **取操作数（FO）：** 从存储器中取操作数(若操作数在寄存器中，则无需此阶段)。
- **执行指令（EI）：**执行指令所需的操作，并将结果存于目的位置(寄存器中)。
- **写操作数（WO）：**将结果存入存储器。

31、流水线的“相关”

- **结构相关：**当指令在重叠执行的过程中，硬件资源满足不了指令重叠执行的要求，发生资源冲突时将产生结构相关。
- **数据相关：**当一条指令需要用到前面指令的执行结果，而这些指令均在流水线中重叠执行时，就可能引起数据相关。
- **控制相关：**当流水线遇到分支指令和其他会改变 PC 值的指令时，会发生控制相关。

32、虚拟存储器是什么？

程序的互斥性原理：两个或两个以上的进程，不能同时进入关于同一组共享变量的临界区域，否则可能发生与时间有关的错误，这种现象被称作进程互斥。也就是说，一个进程正在访问临界资源，另一个要访问该资源的进程必须等待。

在多道程序环境下，存在着**临界资源**，它是指多进程存在时必须互斥访问的资源。也就是某一时刻不允许多个进程同时访问，只能单个进程的访问。我们把这些程序的片段称作**临界区或临界段**，它存在的目的是有效的防止竞争条件又能保证最大化使用共享数据。

程序的局部性原理：是指程序在执行时呈现出局部性规律，即在一段时间内，整个程序的执行仅限于程序中的某一部分。相应地，执行所访问的存储空间也局限于某个内存区域。

局部性原理又表现为：**时间局部性和空间局部性**。

时间局部性是指如果程序中的某条指令一旦执行，则不久之后该指令可能再次被执行；如果某数据被访问，则不久之后该数据可能再次被访问。

空间局部性是指一旦程序访问了某个存储单元，则不久之后。其附近的存储单元也将被访问。

根据程序执行的**互斥性**和**局部性**两个特点，我们允许作业装入的时候只装入一部分，另一部分放在磁盘上，当需要的时候再装入到主存，这样以来，在一个小的主存空间就可以运行一个比它大的作业。

同时，用户编程的时候也摆脱了一定要编写小于主存容量的作业的限制。也就是说，用户的逻辑地址空间可以比主存的绝对地址空间要大。对用户来说，好像计算机系统具有一个容量很大的主存储器，称为“**虚拟存储器**”。

虚存容量不是无限的，虚拟内存的最大容量是由计算机的地址结构确定的。且虚拟内存只是在逻辑上对内存容量加以扩充，其逻辑容量由内存容量和外存容量之和决定。地址总线结构能决定主存的访问范围。所以即使你的磁盘再大，但是地址寄存器访问主存的范围很小的话，虚拟存储器容量依然是很小的。

3.3.4 操作系统

1. 什么是操作系统？

操作系统（Operating System，简称 OS）是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。操作系统是用户和计算机的接口，同时也是计算机硬件和其他软件的接口。操作系统的功能包括管理计算机系统的硬件、软件及数据资源，控制程序运行，改善人机界面，为其它应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，提供了各种形式的用户界面，使用户有一个好的工作环境，为其它软件的开发提供必要的服务和相应的接

口。

2. 操作系统的功能

从资源管理的角度——五大基本功能

1.进程和线程的管理 ——进程线程的状态、控制、同步互斥、通信调度等

2.存储管理——分配/回收、地址转换、存储保护等

3.文件管理——文件目录、文件操作、磁盘空间、文件存取控制

4.设备管理——设备驱动、分配回收、缓冲技术等

5.用户接口——系统命令、编程接口

3. 从电脑开机开始，是如何触发操作系统开始工作的

给主机和其他设备供电——>CPU 读取指令跳转到系统 BIOS 中真正的启动代码处——>硬件自检——>BIOS 把控制权转交给下一阶段的启动程序——>BIOS 按照"启动顺序"，把控制权转交给排在第一位的储存设备——>硬盘启动——>控制权转交给操作系统后，操作系统的内核首先被载入内存。

以 Linux 系统为例，先载入/boot 目录下面的 kernel。内核加载成功后，第一个运行的程序是/sbin/init。它根据配置文件（Debian 系统是/etc/initab）产生 init 进程。这是 Linux 启动后的第一个进程，pid 进程编号为 1，其他进程都是它的后代。

然后，init 线程加载系统的各个模块，比如窗口程序和网络程序，直至执行/bin/login 程序，跳出登录界面，等待用户输入用户名和密码。

4. 论述操作系统中系统调用过程

系统调用提供了用户程序和操作系统之间的接口，应用程序通过系统调用实现其余 OS 的通信，并取得它的服务。系统调用不仅可供所有的应用程序使用，而且也可供 OS 本身的其它部分，如命令处理程序。

系统调用的处理步骤（三步）：

首先，将处理机状态由用户态转为系统态；然后由硬件和内核程序进行系统调用的一般性处理，即首先保护被中断进程的 CPU 环境，将处理机状态字 PSW、程序计数器 PC、系统调用号、用户栈指针以及通用寄存器内容等压入堆栈；再然后将用户定义的参数传送到指定的地址保存起来。

其次，分析系统调用类型，转入相应的系统调用处理子程序。（通过查找系统调用入口表，找到相应处理子程序的入口地址转而去执行它。）

最后，在系统调用处理子程序执行完后，应恢复被中断的货设置新进程的 CPU 现场，然后返回被中断进程或新进程，继续往下执行。

5. 死锁的定义（本人复试被问过）

死锁： 是指两个或两个以上的进程在执行过程中， 因争夺资源而造成的一种互相等待的现象， 若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

6. 硬中断和软中断是什么，区别是什么

软中断：

1、 编程异常通常叫做软中断。

2、 软中断是通讯进程之间用来模拟硬中断的一种信号通讯方式。

3、 中断源发中断请求或软中断信号后,CPU 或接收进程在适当的时机自动进行中断处理或完成软中断信号对应的功能。

4、软中断是软件实现的中断,也就是程序运行时其他程序对它的中断;而硬中断是硬件实现的中断,是程序运行时设备对它的中断。

硬中断：

1、 硬中断是由外部事件引起的因此具有随机性和突发性；软中断是执行中断指令产生的，无外部施加中断请求信号，因此中断的发生不是随机的而是由程序安排好的。

2、 硬中断的中断响应周期，CPU 需要发中断回合信号（ NMI 不需要），软中断的中断响应周期，CPU 不需发中断回合信号。

3、 硬中断的中断号是由中断控制器提供的（ NMI 硬中断中断号系统指定为 02H）；软中断的中断令直接给出，无需使用中断控制器。

4、 硬中断是可屏蔽的（ NMI 硬中断不可屏蔽），软中断不可屏蔽。

区别：

1、 软中断发生的时间是由程序控制的,而硬中断发生的时间是随机的

2、 软中断是由程序调用发生的,而硬中断是由外设引发的

3、 硬件中断处理程序要确保它能快速地完成它的任务,这样程序执行时才不会等待较长时间

7. 描述中断全过程

中断全过程指的是从中断源发出中断请求开始，CPU 响应这个请求，现行程序被中断，转至中断服务程序，直到中断服务程序执行完毕，CPU 再返回原来的程序继续执行的整个过程。大体上可以把中断全过程分为 5 个阶段：中断请求、中断判优、中断响应、中断处理和中断返回。中断处理过程基本上由 3 部分组成，第一部分为准备部分，其基本功能是保护现场，对于非向量中断方式则需要确定中断源，最后开放中断，允许更高级的中断请求打断低级的中断服务程序；第二部分为处理部分，即真正执行具体的为某个中断源服务的中断服务程序；第三部分为结尾部分，首先要关中断，以防止在恢复现场过程中被新的中断请求打断，接着恢复现场，然后开放中断，以便返回原来的程序后可响应其他的中断请求。中断服务程序的最后一条指令一定是中断返回指令。

8. 简述 open 打开文件的过程

首先,操作系统根据文件名 a，在系统文件打开表中查找

第一种情况：

如果文件 a 已经打开，则在进程文件打开表中为文件 a 分配一个表项，然后将该表项的指针指向系统文件打开表中和文件 a 对应的一项;然后再 PCB 中为文件分配一个文件描述符 fd，作为进程文件打开表项的指针,文件打开完成。

第二种情况：

如果文件 a 没有打开，查看含有文件 a 信息的目录项是否在内存中，如果不在，将目录表装入到内存中，作为 cache；根据目录表中文件 a 对应项找到 FCB 在磁盘中的位置；将文件 a 的 FCB 装入到内存中的 Active inode 中；然后在系统文件打开表中为文件 a 增加新的一个表项，将表项的指针指向 Active Inode 中文件 a 的 FCB；然后在进程的文件打开表中分配新的一项，将该表项的指针指向系统文件打开表中文件 a 对应的表项；然后在 PCB 中，为文件 a 分配一个文件描述符 fd，作为进程文件打开表项的指针,文件打开完成。

9. 进程与线程的区别

1、调度：线程是独立调度的基本单位，进程是资源拥有的基本单位。在同一进程中，线程的切换不会引起进程切换。在不同进程中进行线程切换，将会引起进程切换。

2、拥有资源：进程是拥有资源的基本单位，而线程不拥有系统资源（除了少量资源，比如栈，程序计数器，寄存器），不过线程可以访问其隶属进程的系统资源。

3、并发性：在引入线程的操作系统中，不仅进程之间可以并发

执行，而且同一个进程内的多个线程之间也可以并发执行，能提高系统的吞吐量，系统的并发性也更好。

4、系统开销：在创建进程和撤销进程时，系统都要为之分配或回收资源，所以操作系统为进程付出的系统开销远大于创建线程或撤销线程的开销。

5、同步和通信：多线程之间的同步和通信容易实现。

10. 进程和程序的区别

（1）进程是一个动态概念，而程序是一个静态概念。

（2）进程具有并行特征，而程序不反映执行所以没有并行特征

（3）进程是竞争计算机系统资源的基本单位，而程序不反映执行也就不会竞争计算机系统资源

（4）不同的进程可以包含同一程序，只要该程序所对应的数据集不同。

11. 常用的存储保护方法

（1）界限寄存器

上下界寄存器方法

基址、限长寄存器方法

（2）存储保护键：给每个存储块分配一个单独的存储键，它相当于一把锁。

12. 什么是饥饿？与死锁有什么差别？

等待时间给进程推进和响应带来明显影响时成为进程饥饿。

饥饿并不代表系统一点死锁，但至少有一个程序的执行被无限期地推迟。

差别：

1、进入饥饿的进程可以只有一个，但是死锁必须大于等于两个；

2、出于饥饿状态的进程可以是一个就绪进程，但是死锁状态的进程必定是阻塞进程。

13. 操作系统中的信号量

信号量是一个确定的二元组（s，q），其中 s 是一个具有非负初值的整型变量，q 是一个初始状态为空的队列。整型变量 s 表示系统中某类资源的数目，当其值大于 0 时，表示系统中当前可用资源的数目；当其值小于 0 时，其绝对值表示系统中因请求该类资源而被阻塞的进程数目。

信号量分类：

1、整型信号量：所谓整型信号量就是一个用于表示资源个数的整型量

2、记录型信号量（资源信号量）：就是用一个结构体实现，里面包含了表示资源个数的整型量和一个等待队列。

信号量的应用：

1、实现进程同步

2、实现进程互斥

14. 存储器管理应具有的功能

存储管理的主要任务是为多道程序的运行提供良好的环境，方便用户使用存储器，提高存储器的利用率以及从逻辑上扩充存储器，故应具有以下功能：

1) 内存的分配和回收：实施内存的分配，回收系统或用户释放的内存空间。

2) 地址变换：提供地址变换功能，将逻辑地址转换成物理地址。

3) 扩充内存：借助于虚拟存储技术活其他自动覆盖技术，为用户提供比内存空间大的地址空间，从逻辑上扩充内存。

4) 存储保护：保证进入内存的各道作业都在自己的存储空间内运行，互不干扰。

15. 操作系统中的磁盘调度算法

磁盘调度算法目的：使磁盘的平均寻道时间最少。

调度算法	算法思想	优点	缺点
先来先服务算法 FCFS	按照进程请求访问磁盘的先后顺序进行调度。	简单，公平。	未对寻道进行优化，平均寻道时间较长，仅适用于磁盘请求较少的场合。
最短寻道时间优先算法 SSTF	选择与当前磁头所在磁道距离最近的请求作为下一次服务的对象。	较 FCFS 有较好的寻道性能以及较少的寻道时间。	会导致饥饿现象
扫描（电梯调度）算法 SCAN	在磁头当前移动方向上选择与当前磁头所在磁道距离最近的请求最为下一次服务的对象。	具有较好的寻道性能，而且防止了饥饿现象。	存在一个请求刚好被错过而需要等待很久的情形。

循环扫描算法 CSCAN	规定磁头单向移动，如自里向外移动，当磁头移动到最外的磁道时立即返回到最里磁道，如此循环进行扫描。	兼顾较好的寻道性能，防止饥饿现象，同时解决了一个请求等待时间过长的问题。	可能出现磁臂长时间停留在某处不懂的情况（磁臂黏着）。
N-Step-SCAN 算法，对 SCAN 算法的优化。	将磁盘请求队列分成若干个长度为 N 的子队列，磁盘调度将按照 FCFS 依次处理这些子队列，而每处理一个队列时又是按照 SCAN 算法，对一个队列处理后再处理其他队列，将新请求队列放入新队列。	无磁臂黏着。	
FSCAN 算法，对 SCAN 算法的优化。	将请求队列分成两个子队列，将新出现请求磁盘 IO 的进程放入另一个子队列。	无磁臂黏着。	

16. 虚拟存储器，以及相关算法。

基于局部性原理，应用程序在运行之前，仅将那些当前要运行的少数页面或段先装入内存便可运行，其余部分暂时留在盘上。程序运行时，如果它要访问的页已调入内存，便可继续执行下去；但如果程序要访问的页或段尚未调入内存（即缺页），此时程序应利用请求调入功能将它们调入内存，以使程序能继续执行下去。如果此时内存已满，无法装入新的页或段，则需要利用页面置换功能，将内存中暂不使用的页面或段调至盘上，腾出空间用于页面调入内存，是程序继续执行下去。这样，就实现了大的用户程序能在较小的内存空间里运行，也可以在内存中同时装入更多的进程使它们并发运行。从用户角度出发，该系统的内存容量比实际内存容量大很多，故成这样的存储器为虚拟存储器。

相关算法：

页面置换算法

① 最佳置换算法（OPT）：在预知一个进程的页面号引用串的情况下，每次都淘汰以后不再使用的货以后最迟再被使用的页面。该算法不能实现，只能作为一个标准来衡量其他置换算法的优劣。

② 先进先出算法（FIFO）：每次总是淘汰最先进入内存的

页面，也就是将在内存中驻留时间最长的页面淘汰。(可能会产生 Belady 异常，缺页次数随着分配的物理块的增加而增加)。

③ 最近最少使用算法（LRU）：选择最近最少未被使用的页面淘汰，其思想是用以前的页面引用情况来预测将来会出现的页面引用情况。利用了局部性原理。

④ 时钟置换算法（CLOCK）：是 LRU 和 FIFO 的折中，具体方法略。

⑤ 工作集算法

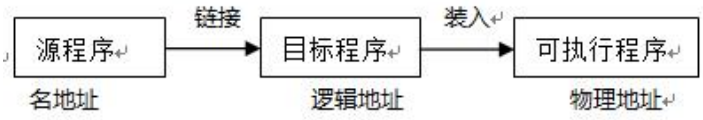
⑥ 工作集时钟算法

⑦ 第二次机会算法

⑧ 最近未使用（NRU）

17. 程序的装入方式有哪些？

补充：应用程序从用户编写的源文件到内内存中执行的进程大致分为三个阶段，经过编译程序将源代码便以为若干个目标模块，在通过链接程序将编译好的目标模块以及所需的库函数链接到一起，形成完整的装入模块，最后通过装入程序将这些装入模块装入内存并执行。（编译，链接，装入）



装入方式：

1）绝对装入：在编译时就知道程序将要驻留在内存的物理地址，编译程序产生含有物理地址的目标代码，不适合多道程序设计。

2）可重定位装入：根据内存当前情况，将装入模块装入到内存的适当位置，地址变换通常在装入时一次完成，之后不再改变，也称静态重定位。当操作系统为程序分配一个以某地址为起始地址的连续主存区域后，重定位时将程序中指令或操作数的逻辑地址加上这个起始地址就得到了物理地址。

3）动态运行装入：允许程序运行时在内存中移动位置，把装入模块装入到内存后的所有地址都是相对地址，在程序执行过程中每当访问到相应指令或数据时，才将要访问的程序或数据的相对地址转换为物理地址。动态重定位的实现要依靠硬件地址变换机构。

18. 程序的链接方式有哪些？

① 静态链接：在程序运行之前，先把各个目标模块及所需库链接为一个完整的可执行程序，以后不再拆开。

② 装入时动态链接：将应用程序编译后所得到的一组目标

模块在装入内存时采用边装入边链接的链接方式。

③ 运行时动态链接：知道程序运行过程中需要一些模块时，才对这些模块进行链接。

19. 交换技术，覆盖技术，以及两者的区别。

覆盖技术：把一个大的程序划分为一系列覆盖，每个覆盖是一个相对独立的程序单位，把程序执行时并不要求同时装入内存的覆盖组成一组，成为覆盖段，这个覆盖段分配到同一个存储区域，这个存储区域成为覆盖区，它与覆盖段一一对应。覆盖段的大小由覆盖段中最大的覆盖来确定。（为了解决内存容量太小的问题，打破了必须将一个程序全部信息装入内存后才能运行的限制）

交换技术：把暂时不用的某个程序及数据部分从内存移到外存中去，以便腾出必要的内存空间；或者把指定的程序或数据从外存读到相应的内存中，并将控制权交给他，让其在系统上运行的一种内存扩充技术。处理器的中级调度就是采用交换技术。

区别：

① 与覆盖技术相比，交换技术不要求程序员给出的程序段之间的覆盖结构；

② 交换技术主要在进程和作业之间进行，覆盖技术主要在同一个进程或作业中进行；

③ 覆盖技术只能覆盖于覆盖程序段无关的程序段，交换进程由换出和换入两个过程组成。

20. 是什么进程线程树图

进程树是一个形象化的比喻，比如一个进程启动了一个程序，而启动的这个进程就是原来那个进程的子进程，依此形成的一种树形的结构，我们可以在进程管理器选择结束进程树，就可以结束其子进程和派生的子进程。

21. 什么是原子操作？什么是 PV 操作？

原子操作是指不会被线程调度机制打断的操作；这种操作一旦开始，就一直运行到结束，中间不会有任何 context switch 。P、V 操作以原语形式实现，保证了对信号量进行操作过程中不会被打断或阻塞。P 操作相当于申请资源，V 操作相当于释放资源。P 操作和 V 操作必定成对出现，但未必在同一个进程中。

22. 硬实时和软实时

硬实时操作系统中，系统必须在特定的时间内完成指定的应用，

具有较强的“ 刚性”，而分时操作系统则注重将系统资源平均地分配给各个应用，不太在意各个应用的进度如何，什么时间能够完成。不过，就算是实时操作系统，其“ 刚性” 和“ 柔性” 的程度也有所不同，就好像是系统的“ 硬度” 有所不同，因而有了所谓的“ 硬实时(hardreal-time)” 和“ 软实时 (softreal-time)”。

硬实时系统有一个刚性的、 不可改变的时间限制， 它不允许任何超出时限的错误。 超时错误会带来损害甚至导致系统失败、或者导致系统不能实现它的预期目标。

软实时系统的时限是一个柔性灵活的， 它可以容忍偶然的超时错误。 失败造成的后果并不严重， 例如在网络中仅仅是轻微地降低了系统的吞吐量。

23. 试说明 DMA 的工作流程。

以从磁盘读入数据为例，说明 DMA 的工作流程。当 CPU 要从磁盘读入数据块时，先向磁盘控制器发送一条读命令。 该命令被送到命令寄存器 CR 中。同时还发送本次要读入数据的内存起始目标地址，送入内存地址寄存器 MAR；本次要读数据的字节数送入数据计数器 DC，将磁 盘中的源地址直接送 DMA 控制器的 I/O 控制逻辑上。然后启动 DMA 控制器传送数据，以后 CPU 便处理其它任务。整个数据传送过程由 DMA 控制器控制。

24. 为何要引入设备独立性？如何实现设备独立性？

现代操作系统为了提高系统的可适应性和可扩展性，都实现了设备独立性或设备无关性。基本含义是应用程序独立于具 体使用的物理设备，应用程序以逻辑设备名请求使用某类设备。实现了设备独立性功能可带来两方面的好处：（1）设备分配时的灵活性；（2）易于实现 I/O 重定向。 为了实现设备的独立性，应引入逻辑设备和物理设备概念。在应用程序中，使用逻辑设备名请求使用某类设备；系统执行时 是使用物理设备名。鉴于驱动程序是与硬件或设备紧密相关的软件，必须在驱动程序之上设置一层设备独立性软件，执行所有设备的公有操作、完成逻辑设备名到物 理设备名的转换（为此应设置一张逻辑设备表）并向用户层（或文件层）软件提供统一接口，从而实现设备的独立性。

25. 何谓提前读、延迟写和虚拟盘？

提前读是指在读当前盘块的同时，将下一个可能要访问的盘块数据读入缓冲区，以便需 要时直接从缓冲区中读取，无需启动磁盘。延迟写是指在写盘块时，将对应缓冲区中的立即写数据暂时不立即写以

备不久之后再被访问，只将它置上“延迟写”标志并 挂到空闲缓冲队列的末尾。当移到空闲缓冲队首并被分配出去时，才写缓冲区中的数据。只要延迟写块仍在空闲缓冲队列中，任何要求访问都可直接从其中读出数据 或将数据写入其中，而不必去访问磁盘。 虚拟盘又称RAM 盘，是利用内存空间仿真磁盘。其设备驱动程序可以接受所有标准的磁盘操作，但这些操作不是在磁盘上而是在内存中，因此速度更快。

26. Windows 下的内存是如何管理的？

Windows 提供了 3 种方法来进行内存管理：虚拟内存，最适合用来管理大型对象或者结构数组；内存映射文件，最适合用来管理大型数据流（通常来自文件）以及在单个计算机上运行多个进程之间共享数据；内存堆栈，最适合用来管理大量的小对象。

Windows 操纵内存可以分两个层面：物理内存和虚拟内存。

其中物理内存由系统管理，不允许应用程序直接访问，应用程序可见的只有一个 2G 地址空间，而内存分配是通过堆进行的。对于每个进程都有自己的默认堆，当一个堆创建后，就通过虚拟内存操作保留了相应大小的地址块（不占有实际的内存，系统消耗很小）。当在堆上分配一块内存时，系统在堆的地址表里找到一个空闲块（如果找不到，且堆创建属性是可扩充的，则扩充堆大小），为这个空闲块所包含的所有内存页提交物理对象（在物理内存上或硬盘的交换文件上），这时就可以访问这部分地址。提交时，系统将对所有进程的内存统一调配，如果物理内存不够，系统试图把一部分进程暂时不访问的页放入交换文件，以腾出部分物理内存。释放内存时，只在堆中将所在的页解除提交（相应的物理对象被解除），继续保留地址空间。

如果要知道某个地址是否被占用/可不可以访问，只要查询此地址的虚拟内存状态即可。如果是提交，则可以访问。如果仅仅保留，或没保留，则产生一个软件异常。此外，有些内存页可以设置各种属性。如果是只读，向内存写也会产生软件异常。

27. 什么是临界区？如何解决冲突？

每个进程中访问临界资源的那段程序称为临界区，每次只准许一个进程进入临界区，进入后不允许其他进程进入。

（1）如果有若干进程要求进入空闲的临界区，一次仅允许一个进程进入；

（2）任何时候，处于临界区内的进程不可多于一个。如已有进程进入自己的临界区，则其它所有试图进入临界区的进程必须等待；

（3）进入临界区的进程要在有限时间内退出，以便其它进程能及时进入自己的临界区；

（4）如果进程不能进入自己的临界区，则应让出 CPU，避免进程出现“忙等”现象。

28. 什么是缓冲区溢出？有什么危害？其原因是什么？

缓冲区溢出是指当计算机向缓冲区内填充数据时超过了缓冲区本身的容量，溢出的数据覆盖在合法数据上。

危害：在当前网络与分布式系统安全中，被广泛利用的 50%以上都是缓冲区溢出，其中最著名的例子是 1988 年利用 fingerd 漏洞的蠕虫。而缓冲区溢出中，最为危险的是堆栈溢出，因为入侵者可以利用堆栈溢出，在函数返回时改变返回程序的地址，让其跳转到任意地址，带来的危害一种是程序崩溃导致拒绝服务，另外一种就是跳转并且执行一段恶意代码，比如得到 shell，然后为所欲为。通过往程序的缓冲区写超出其长度的内容，造成缓冲区的溢出，从而破坏程序的堆栈，使程序转而执行其它指令，以达到攻击的目的。

造成缓冲区溢出的主原因是程序中没有仔细检查用户输入的参数。

29. 进程通信的方式

进程间通信主要包括管道, 系统 IPC(包括消息队列,信号量,共享存储), SOCKET。

管道包括三种:1)普通管道 PIPE, 通常有种限制,一是半双工,只能单向传输;二是只能在父子进程间使用. 2)流管道 s_pipe: 去除了第一种限制,可以双向传输. 3)命名管道:name_pipe,去除了第二种限制,可以在许多并不相关的进程之间进行通讯.

系统 IPC 的三种方式类同,都是使用了内核里的标识符来识别.

管道(pipe): 管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

有名管道 (namedpipe) ： 有名管道也是半双工的通信方式，但是它允许无亲缘关系进程间的通信。

信号量(semophore) ： 信号量是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。

消息队列(messagequeue) ： 消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、

管道只能承载无格式字节流以及缓冲区大小受限等缺点。

信号 (sinal) ： 信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。

共享内存(sharedmemory) ：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量，配合使用，来实现进程间的同步和通信。

套接字(socket) ： 套解口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同机器间的进程通信。

30. 线程的实现方式

线程的实现可分为两大类，用户级线程（user-levelthread，ULT）和内核级线程（kernel-levelthread，KLT）。后者又称为内核支持的线程或轻量级进程。

（1）用户级线程

在一个纯粹的用户级线程软件中，有关线程管理的所有工作都由应用程序完成，内核意识不到线程的存在。

使用用户级线程而不是内核级线程的优点有：

- 线程切换不需要内核态特权，进程并不需要为了线程管理而切换到内核态
- 可以为应用程序量身定做调度算法而不扰乱底层的操作系统调度程序
- 用户级线程可以在任何操作系统中运行，不需要对底层内核进行修改以支持用户级线程

相比内核级线程，用户级线程有两个明显的缺点：

- 许多系统调用都会引起阻塞，当用户级线程执行一个系统调用时，不仅这个线程会被阻塞，进程中的所有线程都会被阻塞
- 在纯粹的用户级线程策略中，一个多线程应用程序不能利用多处理技术

（2）内核级线程

在一个纯粹的内核级线程软件中，有关线程管理的所有工作都是由内核完成的，应用程序部分没有进行线程管理的代码，只有一个到内核线程设施的应用程序编程接口（API）。

该方法克服了用户级线程方法的两个基本缺陷：内核可以同时把同一个进程的多个线程调度到多个处理器中；如果进程中的一个线程被阻塞，内核可以调度同一个进程中的另一个线程。相比用户级线程

它的主要缺点是：把控制从一个线程传送到进程中的另一个线程时，需要到内核的状态切换。

某些操作系统采用组合用户级线程和内核级线程的方法，同一个应用程序中的多个线程可以在多个处理器上并行的运行，某个会引起阻塞的系统调用不会阻塞整个进程。如果设计正确，该方法会结合两种线程的优点，同时减少他们的缺点。

31. 说说分段和分页

页是信息的物理单位，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率；或者说，分页仅仅是由于系统管理的需要，而不是用户的需要。段是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能更好的满足用户的需要。

页的大小固定且由系统确定，把逻辑地址划分为页号和页内地址两部分，是由机器硬件实现的，因而一个系统只能有一种大小的页面。段的长度却不固定，决定于用户所编写的程序，通常由编辑程序在对源程序进行编辑时，根据信息的性质来划分。

分页的作业地址空间是一维的，即单一的线性空间，程序员只须利用一个记忆符，即可表示一地址。分段的作业地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

32. 说出你所知道的保持进程同步的方法?

进程间同步的主要方法有原子操作、信号量机制、自旋锁、管程、会合、分布式系统等。

33. 批处理操作系统、分时操作系统和实时操作系统的特点各是什么？

(1) 批处理操作系统的特点：成批处理，系统吞吐量高，资源利用率高，用户不能直接干预作业的执行。
(2) 分时操作系统的特点：多路性、独立性、及时性、交互性。
(3) 实时操作系统的特点：及时响应、快速处理；高可靠性和安全性；不要求系统资源利用率。

34. 中断和轮询的特点

对 I/O 设备的程序轮询的方式，是早期的计算机系统对 I/O 设备的一种管理方式。它定时对各种设备轮流询问一遍有无处理要求。轮流询问之后，有要求的，则加以处理。在处理 I/O 设备的要求之后，处理机返回继续工作。尽管轮询需要时间，但轮询要比 I/O 设备的速度要快得多，所以一般不会发生不能及时处理的问题。当然，再快的

处理机，能处理的输入输出设备的数量也是有一定限度的。而且，程序轮询毕竟占据了 CPU 相当一部分处理时间，因此，程序轮询是一种效率较低的方式，在现代计算机系统中已很少应用。

程序中断通常简称中断，是指 CPU 在正常运行程序的过程中，由于预先安排或发生了各种随机的内部或外部事件，使 CPU 中断正在运行的程序，而转到为响应的服务程序去处理。

轮询——效率低，等待时间很长，CPU 利用率不高。

中断——容易遗漏一些问题，CPU 利用率高。

35. 什么是 TLB?

TLB 的作用是在处理器访问内存数据的时候做地址转换。TLB 的全称是 Translation Lookaside Buffer，可以翻译做旁路缓冲，是一个具有并行查询能力的特殊高速缓冲寄存器。TLB 中存放了一些页表文件，文件中记录了虚拟地址和物理地址的映射关系。当应用程序访问一个虚拟地址的时候，会从 TLB 中查询出对应的物理地址，然后访问物理地址。TLB 通常是一个分层结构，使用与 Cache 类似的原理。处理器使用一定的算法把最常用的页表放在最先访问的层次。

36. 内存连续分配管理方式有哪些？

① 单一连续分配（静态分配）
② 固定分区分配（分区大小可以不等，但事先必须确定，运行时不能改变）
③ 动态分区分配

37. 动态分区分配的算法有哪些？

- 首次适应算法 First Fit
- 循环首次适应算法 Next Fit
- 最佳适应算法 Best Fit
- 最差适应算法 Worst Fit

38. 内部碎片和外部碎片

内部碎片：分配给作业的存储空间中未被利用的部分。
外部碎片：系统中无法利用的小存储块，比如通过动态内存分配技术从空闲内存区上分配内存后剩下的那部分内存块。

39. 银行家算法

主要思想是避免系统进入不安全状态，在每次进行资源分配时，它首先检查系统是否有足够的资源满足要求，如果有，则先试行分配，并

对分配后的新状态进行安全性检查。如果新状态安全，则正式分配上述资源，否则拒绝分配上述资源。这样就保证系统始终处于安全状态，从而避免死锁现象的发生。

3.3.5 计算机网络

3.3.6 软件工程

一、什么是软件？什么是软件工程？软件工程常见的名词并解释？

软件是一系列按照特定顺序组织的计算机数据和指令的集合。一般来讲软件被划分为系统、应用软件和介于这两者之间的中间件。软件并不只是包括可以在计算机（这里的计算机是指广义的计算机）上运行的电脑程序，与这些电脑程序相关的文档一般也被认为是软件的一部分。简单的说软件就是程序加文档的集合体。

软件也有**其他一些定义**，例如：

- 运行时，能够提供所要求的功能和性能的**指令**或**计算机程序集合**。
- 程序能够满意地处理信息的**数据结构**。
- 描述程序功能需求以及程序如何操作和使用所要求的**文档**。
- 以**开发语言**作为描述语言，可以认为：软件=程序+数据+文档

软件工程中常见的名词解释：

软件危机：是计算机软件的开发和维护过程所遇到的一系列严重的问题。

软件工程方法学包括三个要素：方法、工具和过程。

软件定义时期通常划分成三个阶段：问题定义、可行性研究和需求分析。

软件开发时期通常由四个阶段组成：总体设计、详细设计、编码和单元测试、综合测试。

软件开发方法：用早就定义好的技术集合及符号表示习惯来组织软件生产的过程。主要：**结构化方法**（面向数据流的开发方法，指导思想：自顶向下、逐步求精。基本原则：功能的分解与抽象）、

Jackson 方法（面向数据结构的开发方法）、

维也纳开发方法（VDM）（是一种形式化的开发方法）、

面向对象的开发方法（它有：Booh 方法、Coad 方法、和 OMT 等, UML 语言是面向对象的标准建模语言）。

软件开发的目标：是在规定的投资和时间内，开发出符合用户需求的高质量的软件。

软件工具：一般是指为了支持软件人员开发和维护活动而使用的软件。

软件开发模型：是指软件开发全部过程、活动和任务的结构框架。软件开发模型能清晰、直观地表达软件开发全过程，明确规定了要完成的主要活动和任务，用来作为软件项目工作的基础。

主要有：

- **瀑布模型**（整体开发模型）
- **增量模型**（非整体开发模型）
- **螺旋模型**（是瀑布和增量相结合的模型；一种风险驱动的模型）
- **喷泉模型**（是以用户需求为动力，以对象作为驱动的模型）
- **基于知识的模型**（又称智能模型，是把瀑布模型和专家系统结合在一起的模型）
- **变换模型**（合适于形式化开发的模型）等

模块化：是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。

耦合性：称块间联系。指软件系统结构中各模块间相互联系紧密程度的一种度量。模块之间联系越紧密，其 耦合性就越强，模块的独立性则越差。

内聚性：也称块内联系。指模块的功能强度的度量，即一个模块内部各个元素彼此结合的紧密程度的度量。 模块内元素联系越紧密，内聚性越高。

模块的独立性：指每个模块只完成系统要求的独立的子功能，并且与其他模块的联系最少且接口简单。

概要设计：是在需求分析的基础上通过抽象和分解将系统分解成模块，确定系统功能是实现。

对象：是客观实体在问题域中的抽象。

类：具有相似或相同性质的对象的抽象就是类。

抽象：是认识复杂现象过程中使用的思维工具，即抽出事物本质的共同的特性而暂不考虑它的细节，不考虑 其他因素。

信息隐蔽：指在设计和确定模块时，使得一个模块内包含的信息（过程或数据），对于不需要这些信息的其它 模块来说，是不能访问的。

二、软件开发常见的几种模型，你能说出来几个？请分别解释一下吧？

瀑布模型

瀑布模型是将软件生存周期的各项活动规定为按固定顺序而连

接的若干阶段工作，形如瀑布流水，最终得到软件产品。

瀑布模型是一个项目开发架构，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，那么最好“返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。

瀑布模型有以下**优点：**

1）为项目提供了按阶段划分的检查点。

2）当前一阶段完成后，您只需要去关注后续阶段。

3）可在迭代模型中应用瀑布模型。

增量迭代应用于瀑布模型。第一次迭代解决最大的问题。每次迭代产生一个可运行的版本,同时增加更多的功能。每次迭代必须经过**质量和集成测试**。

4）它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。

瀑布模型有以下**缺点：**

1）各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量。

2）由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发风险。

3）通过过多的强制完成日期和里程碑来跟踪各个项目阶段。

4）瀑布模型的突出缺点是不适应用户需求的变化。

增量模型

增量模型又称为**渐增模型**，也称为**有计划的产品改进模型**，它从一组给定的需求开始，通过构造一系列可执行中间版本来实施开发活动。第一个版本纳入一部分需求，下一个版本纳入更多的需求，依此类推，直到系统完成。每个中间版本都要执行必需的过程、活动和任务。

增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。运用增量模型的软件开发过程是递增式的过程。相对于瀑布模型而言，采用增量模型进行开发，开发人员不需要一次性地把整个软件产品提交给用户，而是可以分批次进行提交。

增量模型的**最大特点**就是将待开发的软件系统**模块化和组件化**。基于这个特点，增量模型具有以下**优点：**

1、将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展。

2、以组件为单位进行开发降低了软件开发的风险。一个开发周

期内的错误不会影响到整个软件系统。

3、开发顺序灵活。开发人员可以对组件的实现顺序进行优先级排序，先完成需求稳定的核心组件。当组件的优先级发生变化时，还能及时地对实现顺序进行调整。

增量模型的**缺点：**

1、要求待开发的软件系统可以被模块化。如果待开发的软件系统很难被模块化，那么将会给增量开发带来很多麻烦。

2、容易退化为边做边改模型，从而使软件过程的控制失去整体性

3、如果增量包之间存在相交的情况且未很好处理，则必须做全盘系统分析

喷泉模型

喷泉模型是一种以**用户需求**为动力，以**对象为驱动**的模型，主要用于描述**面向对象**的软件开发过程。该模型认为软件开发过程**自下而上**周期的各阶段是**相互迭代和无间隙**的特性。

喷泉模型主要用于采用对象技术的软件开发项目。软件的某个部分常常被重复工作多次，相关对象在每次迭代中随之加入渐进的软件成分。**无间隙**指在各项活动之间无明显边界，如分析和设计活动之间没有明显的界限，由于对象概念的引入，表达分析、设计、实现等活动只用对象类和关系，从而可以较为容易地实现活动的**迭代和无间隙**，使其开发自然地包括**复用**。

喷泉模型的**优点：**

- 喷泉模型不像瀑布模型那样，需要分析活动结束后才开始设计活动，设计活动结束后才开始编码活动。
- 该模型的各个阶段没有明显的界限，开发人员可以同步进行开发。
- 可以提高软件项目开发效率，节省开发时间，适应于面向对象的软件开发过程。

喷泉模型的**缺点：**

- 由于喷泉模型在各个开发阶段是重叠的，因此在开发过程中需要大量的开发人员，因此不利于项目的管理。
- 此外这种模型要求严格管理文档，使得审核的难度加大，尤其是面对可能随时加入各种信息、需求与资料的情况。

边做边改模型

当一个软件产品在没有规格说明或主要设计的情况下被开发时，开发者往往不得不重新对产品编码多次直到他们得到正确稳定的产品。这种开发模型就是**边做边改模型**。

边做边改模型**优点：**

在提供给用户使用后，如果程序出现错误，或者用户提出新的要求，开发人员可以重新修改代码，直到用户满意为止。

边做边改模型**缺点：**

边做边改模型的**最重要缺点**是存在于需求。设计和实现中的错误要到整个产品被构建出来后才能被发现。

这种开发模型对任何规模的开发来说都是**不能令人满意**的，其**主要问题**在于：

1）缺少规划和设计环节，软件的结构随着不断的修改越来越糟，导致无法继续修改；

2） 忽略需求环节，给软件开发带来很大的风险；

3） 没有考虑测试和程序的可维护性，也没有任何文档，软件的维护十分困难。

螺旋模型

螺旋模型是一种演化软件开发过程模型，它兼顾了**快速原型**的**迭代**的特征以及**瀑布模型**的**系统化与严格监控**。螺旋模型最大的特点在于引入了其他模型不具备的**风险分析**，使软件在无法排除重大风险时有机会停止，以减小损失。同时，在每个迭代阶段构建原型是螺旋模型用以减小风险的途径。螺旋模型更适合大型的昂贵的系统级的软件应用。

螺旋模型采用一种**周期性**的方法来进行系统开发。这会导致开发出众多的中间版本。该模型是**快速原型法**，以进化的开发方式为中心，在每个项目阶段使用**瀑布模型法**。这种模型的每一个周期都包括需求定义、风险分析、工程实现和评审 4 个阶段，由这 4 个阶段进行迭代。软件开发过程每迭代一次，软件开发又前进一个层次。

螺旋模型强调**风险分析**，使得开发人员和用户对每个演化层出现的风险有所了解，继而做出应有的反应，因此特别适用于庞大、复杂并具有高风险的系统。对于这些系统，风险是软件开发不可忽视且潜在的不利因素，它可能在不同程度上损害软件开发过程，影响软件产品的质量。

螺旋模型**限制条件**

（1）螺旋模型强调风险分析，但要求许多客户接受和相信这种分析，并做出相关反应是不容易的，因此，这种模型往往适应于内部的大规模软件开发。

（2）如果执行风险分析将大大影响项目的利润，那么进行风险分析毫无意义，因此，螺旋模型只适合于大规模软件项目。

（3）软件开发人员应该擅长寻找可能的风险，准确地分析风险，否则将会带来更大的风险

螺旋模型**优点**

1）设计上的灵活性,可以在项目的各个阶段进行变更。

2）以小的分段来构建大型系统,使成本计算变得简单容易。

3）客户始终参与每个阶段的开发,保证了项目不偏离正确方向以及项目的可控性。

4）随着项目推进,客户始终掌握项目的最新信息 ，从而他或她能够和管理层有效地交互。

5）客户认可这种公司内部的开发方式带来的良好的沟通和高质量的产品。

螺旋模型**缺点**

- 很难让用户确信这种演化方法的结果是可以控制的。
- 建设周期长，而软件技术发展比较快，所以经常出现软件开发完毕后，和当前的技术水平有了较大的差距，无法满足当前用户需求。

快速原型模型

原型是指模拟某种产品的原始模型，在其他产业中经常使用。**软件开发中的原型**是软件的一个早期可运行的版本，它反映了**最终系统**的重要特性。

快速原型模型需要迅速建造一个可以运行的软件原型，以便理解和澄清问题，使开发人员与用户达成共识，最终在确定的客户需求基础上开发客户满意的软件产品。 **快速原型模型**允许在**需求分析阶段**对软件的需求进行初步而非完全的分析和定义，快速设计开发出软件系统的原型，该原型向用户展示待开发软件的全部或部分功能和性能；**用户**对该原型进行测试评定，给出具体改进意见以丰富细化软件需求；**开发人员**据此对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护。

快速原型模型又称**原型模型**，它是**增量模型**的另一种形式；它是在开发真实系统之前，构造一个原型，在该原型的基础上，逐渐完成整个系统的开发工作。快速原型模型的**第一步**是建造一个**快速原型**，实现客户或未来的用户与系统的交互，用户或客户对原型进行评价，进一步细化待开发软件的需求。通过逐步调整原型使其满足客户的要求，开发人员可以确定客户的真正需求是什么；**第二步**则在第一步的**基础**上开发客户满意的**软件产品**。

快速原型模型**优点**：

1、克服**瀑布模型**的缺点，减少由于软件需求不明确带来的开发风险。

2、这种模型适合预先不能确切定义需求的软件系统的开发。

快速原型模型**缺点**：

1、所选用的开发技术和工具不一定符合主流的发展

2、快速建立起来的系统结构加上连续的修改可能会导致产品质量低下。

3、使用这个模型的前提是要有一个展示性的产品原型，因此在一定程度上可能会限制开发人员的创新。

三、白盒测试和黑盒测试是什么？

黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的**黑盒测试**意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。

黑盒测试主要是为了发现以下几类错误：

1、是否有不正确或遗漏的功能？

2、在接口上，输入是否能正确的接受？能否输出正确的结果？

3、是否有数据结构错误或外部信息（例如数据文件）访问错误？

4、性能上是否能够满足要求？

5、是否有初始化或终止性错误？

软件的**白盒测试**是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。

白盒测试主要是想对程序模块进行如下检查：

1、对程序模块的所有独立的执行路径至少测试一遍。

2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性。

因此，软件测试有一个致命的缺陷，即测试的不完全、不彻底性。由于任何程序只能进行少量（相对于穷举的巨大数量而言）的有限的测试，在未发现错误前，不能说明程序中没有错误。

四、软件的生命周期是什么？

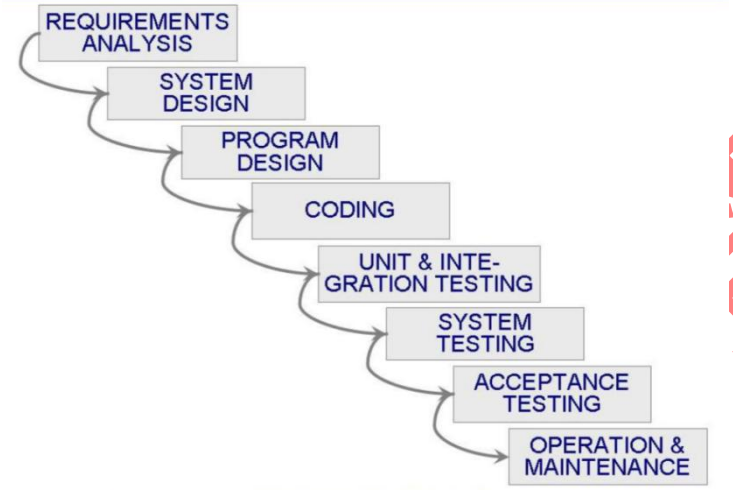
当一个过程涉及到构建一个软件时，这个过程可以被称为软件生命周期。

它包括以下几个阶段：

- 需求分析和定义
- 系统（体系结构）设计
- 程序（详细的/程序化的）设计
- 编写程序（编码/实现）
- 测试：单元、集成、系统
- 系统交付（部署）—维护

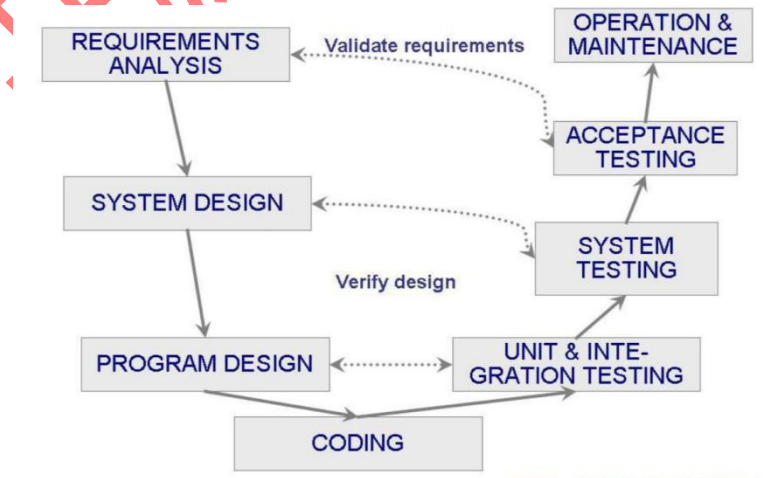
五、软件开发过程模型有哪些？

瀑布模型



- 提议的第一过程开发模型之一
- 服务于很好理解的问题，这些问题在需求上很少或没有变化
- 简单容易的向客户解释
- 它体现了
- —开发过程的非常高层级的看法
- —过程活动的顺序
- 各主要阶段被重要事件和可交付成果（作品）标记

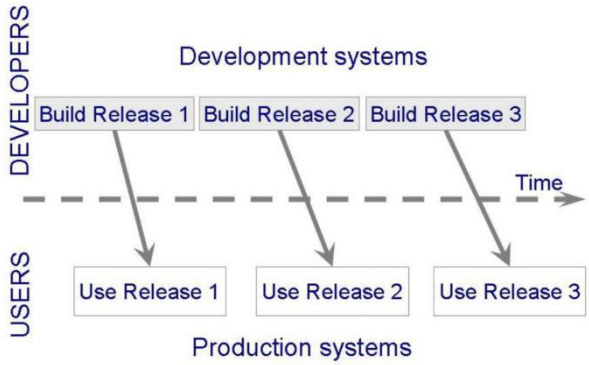
V 模型



- 瀑布模型的一种演变形式

- 使用单元测试来验证程序的设计
- 采用集成测试来验证体系结构（系统）的设计
- 使用验收测试来确认需求
- 如果问题在检验和确认的过程中被发现，在右侧的测试被重新制定之前，V 模型的左侧可以重新执行

阶段性开发：增量和迭代



- 更短的周期时间
- 系统分片交付
- —使得用户拥有一些功能，而其余的正在开发
- 允许两个系统并行运作

—生产系统（发行的版本 n）：目前正在使用

—开发系统（发行的版本 n+1）：下一版本

螺旋模型

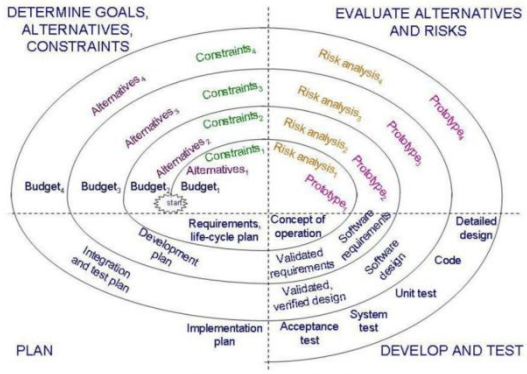


Figure 2.10 the spiral model.

它是瀑布模型和增量模型相结合的模型；一种风险驱动的模型。

六、什么是软件的模块化设计？

模块化，也叫作关注点的分离，它是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。每个模块完成一个特定的子功能，所有的模块按某种方法组装起来，成为一个整体，完成整个系统所要求的功能。

它可以减少错误的发生，同时可以提高软件的可维护性（包容变

化）。

模块化的优点：

- 软件容易被理解和开发
- 程序容易定位错误（因为每个模块都较小）
- 容易更新系统（维护）

七、什么叫耦合？什么叫内聚？为什么说软件开发要高内聚，低耦合

耦合：耦合是软件结构中各模块之间相互连接的一种度量，耦合强弱取决于模块间接口的复杂程度、进入或访问一个模块的点以及通过接口的数据。

内聚：内聚是从功能角度来度量模块内的联系，一个好的内聚模块应当恰好做一件事。它描述的是模块内的功能联系；

软件开发高内聚低耦合？

首先**面向对象**的主要特点是**封装、继承和多态**。实质上是将本来混乱的代码尽可能的抽象出共性，分门别类。最终达到复用，提高开发效率和保证软件质量的目的。

大家都知道蜘蛛以网来捕捉昆虫，因为网中的各个节点是互相联系的，这样一来就增加了蜘蛛网的韧性而不容易被昆虫冲破。而**面向对象**追求的是**代码复用**和**运行稳定**，即在模块之间**减少联系**。可以想象，如果各模块之间联系太过紧密对于代码复用是不容易的，因为很难会再找到同样适合的环境。同时对于运行来说是不利的，软件发生错误也是不可避免的。所以各模块如果联系过于紧密则错一处而动全身，好比多米诺骨牌。所以一个好的软件各模块之间的联系不能过分紧密，即**低耦合**。

面向对象的思想追求的是每个模块的**功能单一**，模块间**越独立越好**。即在划分模块时，只有为了完成同一个功能的各个元素才会被划分到同一个模块中。所以**模块内部**的各个元素必须是**联系紧密**的，否则就说明模块划分还没有到位。所以模块内部需要是联系紧密的，即**高内聚**。

耦合性和内聚性是判断模块化的尺度

内聚性包括：

巧合内聚，逻辑内聚，临时内聚，过程内聚，通信内聚，功能内聚，信息内聚

从左到右是低内聚到高内聚，其中，**功能内聚**是理想化的内聚程度。

一个模块隐藏了数据的表现形式是**信息内聚**。

一个模块隐藏了一个算法是**功能内聚**。

耦合性包括：

内容耦合、公共耦合、控制耦合、标记耦合、数据耦合、无耦合

从左到右是高耦合到低耦合，其中，内容耦合和公共耦合是紧密耦合，标记耦合和数据耦合是松散耦合。

公共耦合—共享数据区和变量名

标记耦合—复杂的数据结构（结构化数据）在模块间传递

数据耦合—只在模块间传递数据值（往往是简单类型）

软件的设计要求是：高内聚，低耦合（松散耦合）

八、什么是程序的接口设计？

接口定义了软件单元对外提供的服务，它明确了模块之间的耦合关系。

接口的五个要素：

- 目的：如函数名
- 先决条件：如接口函数调用前已经做好了哪些准备工作
- 协议：如参数和返回值的类型，指针指向的数据格式
- 后置条件：如返回值、printf 函数接口的效果是在屏幕上显示字符串
- 质量属性：如接口函数的执行时间限制在 1s 内

—函数名、参数和返回值是接口的三个显性的要素。

—前置条件和质量属性是接口的两个隐性要素。

接口的分类：

- 共享数据和变量名（间接/隐含接口）
- call—in functions 函数调用
- call—back functions 函数调用
- 同步调用接口
- 异步调用接口—信号量、消息队列等方式来实现

九、RUP 是什么？

RUP 是**统一软件开发过程**，基于构建开发模型的代表，使用统一建模语言；

RUP 的核心：用例驱动，以体系结构为核心，迭代及增量；

RUP 包括以下几个阶段：

- 先启：定制整个项目范围
- 精化：制定项目计划、功能描述、建立体系结构框架和可执行的“体系结构基线”
- 构建：构造软件产品（实现及测试）
- 产品化：将产品交付给用户

十、敏捷开发是什么？

敏捷开发的**核心观点**是以人为本，提供给客户尽早可交付的软件。

它的**关键**是可交付给客户的产品以及帮助客户实现最大的价值

它的宗旨是：

- 个体与交互　　胜过　　过程和工具
- 可交付的软件　　胜过　　面面俱到的文档
- 客户合作　　胜过　　合同谈判
- 响应变化　　胜过　　遵循计划

十一、什么是极限编程？

极限编程（XP）是一个轻量级的、灵巧的软件开发方法；同时它也是一个非常严谨和周密的方法。它的基础和价值观是交流、朴素、反馈和勇气；

即任何一个软件项目都可以从四个方面入手进行改善：

加强交流；从简单做起；寻求反馈；勇于实事求是。

极限编程中有四个核心价值是我们在开发中必须注意的：沟通、简单、反馈、勇气、此外还扩展了第五个价值观：谦逊。XP 用“沟通、简单、反馈、勇气和谦逊”来减轻开发压力和包袱；XP 用自己的实践，在一定范围内成功地打破了软件工程“必须重量”才能成功的传统观念。

XP 是一种**近螺旋式**的开发方法，它将复杂的开发过程分解为一个个相对比较简单的小周期；通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。

十二、（你做过哪些项目？你从中收获哪些软件工程领域的思想？请说一说）

面向对象思想

(1)面向对象方法

面向对象方法学的出发点和基本原则，是尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界、解决问题的方法与过程，从而使得实现解法的解空间（也称为求解域）与描述问题的问题空间（也称为问题域）在结构上尽可能一致。

面向对象方法不是把程序看做是工作在数据上的完成特定子功能的过程或函数的集合，而是把程序看做是相互协作而又彼此独立的对象的集合。每个对象就像一个微型程序，有自己的数据、操作、功能和目的。这样做就向着减少语义断层的方向迈进一大步，这样的软

件稳定性好，较易开发且容易理解和维护。

概括地说，面向对象方法有下述 **4 个要点：**

①认为客观世界是由各种对象组成的，任何事物都是对象，复杂的对象由比较简单的对象以某种方式组合而成。因此，面向对象的软件系统是由对象组成的，软件中的任何元素都是对象，复杂的软件对象由比较简单的软件对象组合而成。由此可知，面向对象方法用对象分解取代了传统方法的功能分解。

②把所有对象都划分成各种对象类（简称为类），每个类都定义了一组数据和一组方法。数据用于表示对象的静态属性，是对象的状态信息。类中定义的方法是允许施加于该类对象上的操作，用于实现对象的动态行为。

③按照子类（或称为派生类）与父类（或称为基类）的关系，把若干个类组成一个层次结构的系统（也称为类等级）。在类等级中，下层的派生类自动具有上层基类的特性（包括数据和方法），这种现象称为继承。

④对象彼此之间仅能通过传递消息互相联系。对象与传统的数据有本质区别，它不是被动地等待外界对它施加操作，相反，它是进行处理的主体，必须发消息请求它执行它的某个操作，处理它的私有数据，而不能从外界直接对它的私有数据进行操作。

(2)对象的概念

在应用领域中有意义的、与所要解决的问题密切相关的任何事物都可以作为对象，它既可以是对具体的物理实体的抽象，也可以代表人为的概念或任何有明确边界和意义的东西。对象是由描述该对象属性的数据以及可以对这些数据施加的全部操作封装在一起构成的统一体。通常把对象的操作称为服务或方法。

从面向对象程序设计的角度可以把对象定义为：对象是具有相同状态的一组操作的集合。从信息模拟的角度可以把对象定义为：对象是对问题域中某个东西的抽象，这种抽象反映了系统保存有关这个东西的信息或与它交互的能力。也就是说，对象是对属性值和操作的封装。

对象具有下述的一些基本特点。

- 以数据为中心。
- 对象是主动的，是数据处理的主体。
- 实现了数据封装。

- 本质上具有并行性。
- 模块独立性好。

模块化设计思想

模块化设计，简单地说就是程序的编写不是开始就逐条录入计算机语句和指令，而是首先用主程序、子程序、子过程等框架把软件的主要结构和流程描述出来，并定义和调试好各个框架之间的输入、输出链接关系。逐步求精的结果是得到一系列以功能块为单位的算法描述。以功能块为单位进行程序设计，实现其求解算法的方法称为模块化。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

模块化设计是指在对一定范围内的不同功能或相同功能不同性能、不同规格的产品进行功能分析的基础上，划分并设计出一系列功能模块，通过模块的选择和组合可以构成不同的产品，以满足市场的不同需求的设计方法。

模块化设计原则

① 力求以少量的模块组成尽可能多的产品，并在满足要求的基础上使产品精度高、性能稳定、结构简单、成本低廉，模块间的联系尽可能简单；

② 模块的系列化，其目的在于用有限的产品品种和规格来最大限度又经济合理地满足用户的要求。

模块化设计三大特征

模块是模块化设计和制造的功能单元，具有**三大特征：**

1.相对独立性，可以对模块单独进行设计、制造、调试、修改和存储，这便于由不同的专业化企业分别进行生产；

2.互换性，模块接口部位的结构、尺寸和参数标准化，容易实现模块间的互换，从而使模块满足更大数量的不同产品的需求；

3.通用性，有利于实现横系列、纵系列产品间的模块的通用，实现跨系列产品间的模块的通用。

十三、什么是面向对象，封装，继承，多态？

面向对象的方法是运用对象、类、继承、封装、聚合、消息传送、多态性等构造系统的软件的开发方法。

面向对象=对象+类+继承+通信

采用这四个概念开发的软件系统是面向对象的。

面向对象的三个基本特征是：**封装、继承、多态**。

- **封装**是面向对象的特征之一，是对象和类概念的主要特性。

封装，也就是把客观事物封装成抽象的类，并且类可以把自己的数据和方法只让可信的类或者对象操作，对不可信的类进行信息隐藏。

- **继承**是面向对象编程语言的一个主要功能。

继承是指这样一种能力：它可以使用现有类的所有功能，并在无需重新编写原来的类的情况下对这些功能进行扩展。继承可实现代码重用。

通过继承创建的新类称为“子类”或“派生类”。

被继承的类称为“基类”、“父类”或“超类”。

继承的过程，就是从一般到特殊的过程。

要实现继承，可以通过“继承”和“组合”来实现。

- **多态**是指同一个操作作用于不同对象时可以有不同的解释，并产生不同的执行结果。

多态性发生在代码编写中的接口交互方面，但代码行为依赖于对象，这个对象与运行时的接口和对象方法的实现相关联。

对象组合和多态性是一个面向对象设计的重要特征，它们使结果系统在许多方面更有用。

十四、什么是 uml?有哪些工具可以画 uml 图？说一说依赖，耦合，聚合三种的关系。

Unified Modeling Language (UML)又称**统一建模语言**或**标准建模语言**，是始于 1997 年一个 OMG 标准，它是一个支持模型化和软件系统开发的图形化语言，为软件开发的所有阶段提供**模型化**和**可视化**支持，包括由需求分析到规格，到构造和配置。 **面向对象的分析与设计(OOAD)**方法的发展在 80 年代末至 90 年代中出现了一个高潮，**UML** 是这个高潮的产物。它不仅统一了 Booch、Rumbaugh 和 Jacobson 的表示方法，而且对其作了进一步的发展，并最终统一为大众所接受的标准建模语言。

统一建模语言（UML）是**面向对象软件**的标准化建模语言。UML 因其简单、统一的特点，而且能表达软件设计中的**动态和静态信息**，目前已成为**可视化建模语言**的工业标准。在软件无线电系统的开发过程中，统一建模语言可以在整个设计周期中使用，帮助设计者缩短设计时间，减少改进的成本，使软硬件分割最优。

UML 由 3 个**要素**构成：UML 的基本构造块、支配这些构造块如何放置在一起的规则和运用于整个语言的公用机制。

UML 有 3 种**基本的构造块：**事物、关系和图。

事物是对模型中最具有代表性的成分的抽象，包括**结构事物**，如类、接口、协作、用例、主动类、组件和节点；**行为事物**，如交互、状态机、分组事物（包，Package）、注释事物（注解，Note）。

关系用来把事物结合在一起，包括**依赖、关联、泛化和实现**关系。

UML 定义了 **5 类图，10 种模型图**

五种类图定义：

1.**用例图：**从用户角度描述系统功能，并指各功能的操作者。

2.**静态图：**包括类图，包图，对象图。

类图：描述系统中类的静态结构

包图：是包和类组成的，表示包与包之间的关系，包图描述系统的分层结构

对象图：是类图的实例

3.**行为图：**描述系统动态模型和对象组成的交换关系。包括状态图和活动图

活动图：描述了业务实现用例的工作流程

状态图：是描述状态到状态控制流，常用于动态特性建模

4.**交互图：**描述对象之间的交互关系

顺序图：对象之间的动态合作关系，强调对象发送消息的顺序，同时显示对象之间的交互

合作图：描述对象之间的协助关系

5.**实现图：**

配置图：定义系统中软硬件的物理体系结构

UML 提供的**基本模型图**包括：

- 用例图：展示系统外部的各类执行者与系统提供的各种用例之间的关系
- 类图：展示系统中类的静态结构(类是指具有相同属性和行为的对象，类图用来描述系统中各种类之间的静态结构)
- 对象图：是类图的一种实例化图(对象图是对类图的一种实例化)
- 包图：是一种分组机制。在 UML1.1 版本中，包图不再看作一种独立的模型图)
- 状态图：描述一类对象具有的所有可能的状态及其转移关系(它展示对象所具有的所有可能的状态以及特定事件发生时状态的转移情况)
- 顺序图：展示对象之间的一种动态协作关系(一组对象组成，随时间推移对象之间交换消息的过程，突出时间关系)
- 合作图：从另一个角度展示对象之间的动态协作关系(对象间动态协作关系，突出消息收发关系)
- 活动图：展示系统中各种活动的执行流程(各种活动的执行顺序、执行流程)
- 构件图：展示程序代码的物理结构(描述程序代码的组织结构，各种构件之间的依赖关系)
- 配置图：展示软件在硬件环境中(特别是在分布式及网络环

境中)的配置关系(系统中硬件和软件的物理配置情况和系统体系结构)

UML 画图工具：

画 **UML 图**的工具大致可以分为两类，一类是专业的绘图工具，带了画 UML 的功能，如 Visio、Dia；另一类是专门用来制作 UML 图的，如 ArgoUML 和 Rose，通常都有根据 UML 图直接生成代码。

如果只是简单的画下，这几个工具都够用，但是如果对细节要求严格一些，则都有不足的地方。如果以 R.Martin 的《敏捷软件开发》中的 UML 图为标准，则用这几款软件都无法严格做出其中的图形。当然对于 Visio、Dia 这样的绘图工具，还可以一点一点用基本图形拼出来，但这就烦了一点。例如在序列图（sequence diagram）里，一个调用除了有消息名以外，还可以有返回值，用一个带箭头的小圈表示，这个在上面的几个工具里都没有直接提供的。

UML 建模工具 Visio 、Rational Rose、PowerDesign 的比较

ROSE 是直接从 UML 发展而诞生的设计工具，它的出现就是为了对 UML 建模的支持，ROSE 一开始没有对数据库端建模的支持，但是在现在的版本中已经加入数据库建模的功能。ROSE 主要是在开发过程中的各种语义、模块、对象以及流程，状态等描述比较好，主要体现在能够从各个方面和角度来分析和设计，使软件的开发蓝图更清晰，内部结构更加明朗（但是它的结构仅仅对那些对掌握 UML 的开发人员，也就是说对客户了解系统的功能和流程等并不一定很有效），对系统的代码框架生成有很好的支持。但对数据库的开发管理和数据库端的迭代不是很好。

PowerDesigner 原来是对数据库建模而发展起来的一种数据库建模工具。直到 7.0 版才开始对面向对象的开发的支持，后来又引入了对 UML 的支持。但是由于 PowerDesigner 侧重不一样，所以它对数据库建模的支持很好，支持了能够看到的 90%左右的数据库，对 UML 的建模使用到的各种图的支持比较滞后。但是在最近得到加强。所以使用它来进行 UML 开发的并不多，很多人都是用它来作为数据库的建模。如果使用 UML 分析，它的优点是生成代码时对 Sybase 的产品 PowerBuilder 的支持很好（其它 UML 建模工具则没有或者需要一定的插件），其他面向对象语言如 C++，Java，VB,C#等支持也不错。但是它好像继承了 Sybase 公司的一贯传统，对中国的市场不是很看看好，所以对中文的支持总是有这样或那样的问题。

UML 建模工具 **Visio** 原来仅仅是一种画图工具，能够用来描述各种图形（从电路图到房屋结构图），也是到 VISIO2000 才开始引进软件分析设计功能到代码生成的全部功能，它可以说是目前最能够用图形方式来表达各种商业图形用途的工具（对软件开发中的 UML 支

持仅仅是其中很少的一部分）。它跟微软的 office 产品的能够很好兼容。能够把图形直接复制或者内嵌到 WORD 的文档中。但是对于代码的生成更多是支持微软的产品如 VB,VC++,MS SQL Server 等（这也是微软的传统），所以它可以说用于图形语义的描述比较方便，但是用于软件开发过程的迭代开发则有点牵强。

依赖、耦合、聚合的关系

类图可能是 **UML** 中使用的最多的一种图。

类与类之间的**关系**可以根据**关系的强度**依次分为以下五种：

依赖关系---关联关系---聚合关系---组合关系---泛化关系

依赖(Dependency)关系是类与类之间的联接。依赖关系表示一个类依赖于另一个类的定义。例如，一个人(Person)可以买车(car)和房子(House)，Person 类依赖于 Car 类和 House 类的定义，因为 Person 类引用了 Car 和 House。与关联不同的是，Person 类里并没有 Car 和 House 类型的属性，Car 和 House 的实例是以参量的方式传入到 buy() 方法中去的。一般而言，依赖关系在 Java 语言中体现为局域变量、方法的形参，或者对静态方法的调用。

关联(Association）关系是类与类之间的联接，它使一个类知道另一个类的属性和方法。关联可以是双向的，也可以是单向的。在 Java 语言中，关联关系一般使用成员变量来实现。

聚合(Aggregation) 关系是关联关系的一种，是强的关联关系。聚合是整体和个体之间的关系。例如，汽车类与引擎类、轮胎类，以及其它的零件类之间的关系便整体和个体的关系。与关联关系一样，聚合关系也是通过实例变量实现的。但是关联关系所涉及的两个类是处在同一层次上的，而在聚合关系中，两个类是处在不平等层次上的，一个代表整体，另一个代表部分。

组合(Composition) 关系是关联关系的一种，是比聚合关系强的关系。它要求普通的聚合关系中代表整体的对象负责代表部分对象的生命周期，组合关系是不能共享的。代表整体的对象需要负责保持部分对象和存活，在一些情况下将负责代表部分的对象湮灭掉。代表整体的对象可以将代表部分的对象传递给另一个对象，由后者负责此对象的生命周期。换言之，代表部分的对象在每一个时刻只能与一个对象发生组合关系，由后者排他地负责生命周期。部分和整体的生命周期一样。

泛化关系是学术名称，通俗的来讲，它通常表示类与类之间的继承关系，接口与接口之间的继承关系，或类对接口的实现关系。一般

化（泛化）的关系是从子类指向父类的，与**继承或实现的方法相反**。

以上关系的**耦合度**依次**增强**（耦合度可理解为当一个类发生变更时，对其他类造成的影响程度；影响越小则耦合度越弱，影响越大耦合度越强）。由定义我们已经知道，**依赖关系**实际上是一种**比较弱**的关联，**聚合关系**是一种**比较强**的关联，而**组合关系**则是一种**更强**的关联，所以笼统的来区分的话，实际上这四种关系、都是**关联关系**。

依赖关系比较好区分，它是**耦合度最弱**的一种，在 java 中表现为局域变量、方法的形参，或者对静态方法的调用，如下面的例子：Driver 类依赖于 Car 类，Driver 的三个方法分别演示了依赖关系的三种不同形式。

关联关系在 java 中一般使用成员变量来实现，有时也用方法形参的形式实现。依然使用 Driver 和 Car 的例子，使用方法参数形式可以表示依赖关系，也可以表示关联关系，毕竟我们无法在程序中太准确的表达语义。在本例中，使用成员变量表达这个意思：车是我自己的车，我“拥有”这个车。使用方法参数表达：车不是我的，我只是个司机，别人给我什么车我就开什么车，我使用这个车。

聚合关系是是一种比较强的关联关系，java 中一般使用成员变量形式实现。对象之间存在着整体与部分的关系。

关联关系和依赖关系的区别：

1、从类的属性是否增加的角度看：

- 发生**依赖关系**的两个类都不会增加属性。其中的一个类作为另一个类的方法的参数或者返回值，或者是某个方法的变量而已。
- 发生**关联关系**的两个类，其中的一个类成为另一个类的属性，而属性是一种更为紧密的耦合，更为长久的持有关系。

2、从关系的生命周期来看：

- **依赖关系**是仅当类的方法被调用时而产生，伴随着方法的结束而结束了。
- **关联关系**是当类实例化的时候即产生，当类销毁的时候，关系结束。相比依赖讲，关联关系的生存期更长。

聚合关系和组合关系：

例子： 大雁喜欢热闹害怕孤独，所以它们一直过着群居的生活，这样就有了雁群，每一只大雁都有自己的雁群，每个雁群都有好多大雁，大雁与雁群的这种关系就可以称之为**聚合**。另外每只大雁都有两只翅膀，大雁与雁翅的关系就叫做**组合**。

由此可见：

聚合的关系明显没有**组合**紧密，大雁不会因为它们的群主将雁群解散而无法生存；

而雁翅就无法脱离大雁而单独生存——**组合关系**的类具有相同

的生命周期。

这两种关系的**区别是：**

1、构造函数不同

- **聚合类**的构造函数中包含另一个类的实例作为参数；

因为构造函数中传递另一个类的实例，因此大雁类可以脱离雁群类独立存在。

- **组合类**的构造函数包含另一个类的实例化；

因为在构造函数中进行实例化，因此两者紧密耦合在一起，同生同灭，翅膀类不能脱离大雁类存在。

2、信息的封装性不同

- 在**聚合关系**中，客户端可以同时了解 GooseGroup 类和 Goose 类，因为他们是独立的。
- 在**组合关系**中，客户端只认识大雁类，根本不知道翅膀类的存在，因为翅膀类被严密地封装在大雁类中。

十五、什么是结构化分析方法？

结构化分析方法（结构化方法）是一种软件开发方法，一般利用图形表达用户需求，强调开发方法的结构合理性以及所开发软件的结构合理性。

结构是指系统内各个组成要素之间的相互联系、相互作用的框架。结构化开发方法提出了一组提高软件结构合理性的准则，如分解与抽象、模块独立性、信息隐蔽等。针对软件生存周期各个不同的阶段，它有结构化分析（SA）和结构化程序设计（SP）等方法。

十六、说一说什么是数据建模，什么是 ER 图，什么是 DFD 图、什么是数据字典？

数据建模指的是对现实世界各类数据的抽象组织，确定数据库需要管辖的范围、数据的组织形式等直至转化成**现实的数据库**。将经过系统分析后抽象出来的**概念模型**转化为**物理模型**后，在 visio 或 erwin 等工具建立数据库实体以及各实体之间关系的过程（实体一般是表）。

在软件工程中，**数据建模**是运用正式的数据建模技术，建立信息系统的数据模型的过程。

数据建模是一种用于定义和分析数据的要求和其需要的相应支持的信息系统的过程。因此，数据建模的过程中，涉及到的专业数据建模工作，与企业的利益和用户的信息系统密切相关。

从需求到实际的数据库，有三种不同的类型。用于**信息系统**的数据模型作为一个**概念数据模型**，本质上是一组记录数据要求的最初的规范技术。数据首先用于讨论适合企业的最初要求，然后被转变为一

个**逻辑数据模型**，该模型可以在数据库中的数据结构概念模型中实现。一个概念数据模型的实现可能需要多个逻辑数据模型。数据建模中的最后一步是确定从**逻辑数据模型**到**物理数据模型**中再到对数据访问性能和存储的具体要求。数据建模定义的不只是数据元素，也包括它们的结构和它们之间的关系。

数据建模大致分为三个阶段，**概念建模阶段**，**逻辑建模阶段**和**物理建模阶段**。其中概念建模和逻辑建模阶段与数据库厂商毫无关系，换言之，与 MySQL，SQL Server，Oracle 没有关系。物理建模阶段和数据库厂商存在很大的联系，因为不同厂商对同一功能的支持方式不同，如高可用性，读写分离，甚至是索引，分区等。

实际工作中，在**概念建模阶段**，主要做三件事：

1. 客户交流
2. 理解需求
3. 形成实体

这也是一个**迭代**，如果先有需求，尽量去理解需求，明白当前项目或者软件需要完成什么，不明白或者不确定的地方和客户及时交流，和客户二次确认过的需求，落实到实体；

逻辑建模阶段：对实体进行细化，细化成具体的表，同时丰富表结构。这个阶段的**产物**是，可以在数据库中生成具体的表及其他数据库对象（包括，主键，外键，属性列，索引，约束甚至是视图以及存储过程）。

物理建模阶段：这个阶段将在逻辑建模阶段创建的各种数据库对象生成为相应的 SQL 代码，运行来创建相应具体数据库对象（大多数建模工具都可以自动生成 DDL SQL 代码）。但是这个阶段我们不仅仅创建数据库对象，针对业务需求，我们也可能做如数据拆分（水平或垂直拆分），如 B2B 网站，我们可以将商家和一般用户放在同一张表中。甚至是读写分离，这个阶段也会涉及到集群的事情。

数据流图：简称 **DFD**，是 SA(结构化分析)方法中用于表示系统逻辑模型的一种工具，它以图形的方式描绘数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。

数据字典：简称 **DD**，就是用来定义数据流图中的各个成分具体含义的，它以一种准确的、无二义性的说明方式为系统的分析、设计及维护提供了有关元素的一致的定义和详细的描述。它由五类条目组成：数据流、数据项、数据结构、数据存储、数据处理。

结构化分析模型的组成

- 数据建模（ERD 实体关系图）和对象描述（E-R 中对象属性）

- 功能建模和数据流图（DFD 数据流图）
- 基本加工逻辑说明（PSPEC 加工规格说明：描述数据流图的每个功能）
- 行为建模（STD 状态迁移图）
- 数据字典（核心）

十七、结构化分析方法和面向对象方法的区别？

- 从**概念**方面看

结构化是功能的集合，通过模块以及模块与模块之间的分层调用关系实现

OO 是事物的集合，通过对象以及对象和对象之间的通讯联系实现

- 从**构成**方面看

结构化=过程+数据，以过程为中心

OO=（数据+相应操作）的封装以对象为中心

- 从**运行控制**方面看

结构化采用顺序处理方式，由过程驱动控制

OO 采用交互式、并行处理方式，由消息驱动控制

- 从**开发**方面看

结构化的重点是设计

OO 的重点是分析

- 从**应用**方面看

结构化适合数据类型简单的数值计算和数据统计管理软件的开发

OO 适合大型复杂的人机交互软件和数据统计软件的开发

3.3.7 数据库

1、数据库恢复有哪些实现方式？

一，建立冗余数据；

建立冗余数据最常用的技术是数据转储和登记日志文件。通常在一个数据库系统中，这两种方法是一起使用的。

（1）数据转储

转储：DBA 定期地将整个数据库复制到磁带或另一个磁盘上保存起来的过程。这些备用的数据称为后备副本（backup）或后援副本。

A.静态转储：当系统中无运行事务时进行转储，转储开始时数据库处于一致状态，转储期间不允许对数据库的任何存取、修改活动。优点：实现简单。缺点：降低了数据库的可用性——>转储必须等用户事务结束；新的事务必须等转储结束。

B.动态转储：转储操作与用户事务并发进行，转储期间允许对数据库进行存取或修改。优点：不用等待正在进行的用户事务结束；不会影响新事务的运行。

缺点：不能保证副本中的数据正确有效；利用动态转储得到的副本进行故障恢复，①需要把动态转储期间各事务对数据库的修改活动登记下来，建立日志文件；②后备副本加上日志文件才能把数据库恢复到某一时刻的正确状态。

C.海量转储：每次转储全部数据库

D.增量转储：只转储上次转储后更新过的数据。使用海量转储得到的后备副本进行恢复往往更方便如果数据库很大，事务处理又十分频繁，则增量转储方式更实用有效。

（2）登记日志文件（Logging）

日志文件：用于记录事务对数据库的更新操作的文件。用于进行事务故障恢复和系统故障恢复。主要格式：

A.以记录为单位的日志文件。

日志记录：各个事务的开始标记、各个事务的结束标记、各个事务的所有更新操作。每个日志记录的主要内容：事务标识（标明是哪个事务）、操作的类型（插入、删除或修改）、操作对象（记录内部标识）、更新前数据的旧值（对插入操作而言，此项为空值）、更新后数据的新值（对删除操作而言，此项为空值）。

B.以数据块为单位的日志文件。

日志记录：事务标识、被更新的数据块。登记日志文件时必须遵循两条原则（保证数据库是可恢复的）：①登记的次序严格按并发事务执行的时间次序；②写日志文件，后写数据库。操作的类型和操作对象等信息不必放入日志记录中，因为将更新前的整个块和更新后的整个块都放入日志中。

二，利用这些冗余数据实施数据库恢复。

2、各类数据库故障的恢复策略。

"数据故障恢复"和"完整性约束"、"并发控制"一样，都是数据库数据保护机制中的一种完整性控制。要求 DBMS 要有一套故障后的数据恢复机构，保证数据库能够回复到一致的、正确地状态去。所有的数据恢复的方法都基于数据备份。

（1）事务故障：

事务在运行至正常终止点前被终止。

恢复操作：利用日志文件撤销此事务已对数据库进行的修改。由系统自动完成，对用户透明。

系统的恢复步骤是：

①反向扫描日志文件（即从最后向前扫描日志文件），查找该事务的更新操作。

②对该事务的更新操作执行逆操作，即将日志记录中“更新前的值”写入数据库。若记录中是插入操作，则相当于做删除操作；若记录中是删除操作，则相当于做插入操作；若记录中是修改操作，则相当于用修改前值代替修改后值。

③继续反向扫描日志文件，查找该事务的其它更新操作，并作同样处理。

④如此处理下去，直至读到此事务的开始标记。

（2）系统故障：

系统故障造成数据库不一致状态的原因：①未完成事务对数据库的更新可能已写入数据库，②已提交事务对数据库的更新可能还留在缓冲区没来得及写入数据库。

恢复操作：把未完成事务的故障撤销，并重做已完成的事物。系统在重新启动时自动完成，不需要用户干预。

系统的恢复步骤是：

①正向扫描日志文件（即从头扫描日志文件）：把故障发生前已经提交的事务（既有 BEGIN TRANSACTION 记录，也有 COMMIT 记录）的事务标识记入重做（REDO）队列。把故障发生时尚未完成的事务（只有 BEGIN TRANSACTION 记录，无相应的 COMMIT 记录）的事务标识记入撤销队列。

②对撤销队列中的各个事务进行撤销（UNDO）处理：反向扫描日志文件，对每个 UNDO 事务的更新操作执行逆操作，即将日志记录中“更新前的值”写入数据库。

③对重做队列中的各个事务进行重做（REDO）处理。其方法是，正向扫描日志文件，对每个 REDO 事务重新执行日志文件登记的操作，即将日志记录中“更新后的值”写入数据库。

（3）介质故障的恢复

介质故障：磁盘上的物理数据和日志文件被破坏，这是最严重的一种故障。

恢复操作：重装数据库，然后重做已完成的事务。

①装入最新的数据库后备副本（离故障发生时刻最近的转储副本），是数据库恢复到最近一次转储时的一致性状态。对于动态转储的数据库副本，还需同时装入转储开始时刻的日志文件副本，利用恢复系统故障的方法（即 REDO+UNDO），才能将数据库恢复到一致性状态。

②装入相应的日志文件副本（即转储结束时刻的日志文件副本），重做已完成的事务。即首先扫描日志文件，找出故障发生时已提交的事务的标识，将其记入重做队列。然后正向扫描日志文件，对重做队列中的所有事务进行重做处理。即将日志记录中“更新后的值”写入数据库。

3、数据库并发操作可能带来哪些数据的不一致性问题？

事务具有 ACID 特性（原子特性）。事务是并发控制的基本单位，在事务处理过程中需要保证事务 ACID 特性。多个事务对数据库的并发操作可能造成事务 ACID 特性可能遭到破坏。

（1）丢失修改（Lost Update）

当两个或多个事务（或两个或多个用户）选择同一行，然后基于最初选定的值更新该行时，会发生丢失更新问题。每个事务都不知道其它事务的存在（或每个用户操作时并不会考虑同一时刻是否有别的用户进行着同样的操作）。最后的更新将重写由其它事务所做的更新，这将导致数据丢失。

（2）不可重复读（Non-Repeatable Read）

事务 T_1 读取数据后，事务 T_2 执行更新操作，使 T_1 无法再现前一次读取结果。

包括三种情况：

①事务 T_1 读取某一数据后，事务 T_2 对其做了修改，当事务 T_1 再次读该数据时，得到与前一次不同的值。

②事务 T_1 按一定条件从数据库中读取了某些数据记录后，事务 T_2 删除了其中部分记录，当 T_1 再次按相同条件读取数据时，发现某些记录消失了。

③事务 T_1 按一定条件从数据库中读取某些数据记录后，事务 T_2 插入了一些记录，当 T_1 再次按相同条件读取数据时，发现多了一些记录。

后两种不可重复读有时也称为幻影（Phantom Row）现象。

（3）读“脏”数据（Dirty Read）

当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另外一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。

并发控制就是要用正确的方式调度并发操作，使一个用户事务的执行不受其它事务的干扰，避免造成数据的不一致性。有时对数据库的应用允许某些不一致性，例如有些统计工作涉及数据量很大，读到一些“脏”数据对统计精度没什么影响，这时可以降低对一致性的要求以减少系统开销。

4、并发控制的主要技术是什么？

主要技术有封锁（Locking）、时间戳（Timestamp）和乐观控制法。

（1）封锁：事务 T 在对某个数据对象例如表、记录等操作之前，先向系统发出请求，对其加锁。加锁后事务 T 就对该数据对象有了一定的控制，在事务 T 释放它的锁之前，其它的事务不能更新此数据对象。商用的 DBMS 一般都采用封锁方法。

封锁类型：

①排它锁（Exclusive Locks，简称 X 锁），又称写锁。事物 T 在给数据对象 A 加上 X 锁后，其它事务不能再对 A 加锁，只允许 T 对 A 进行都写操作。保证了其它事务在 T 释放 A 上的锁之前不能再读取和修改 A。当 T_1 给 A 加 X 锁时，T_2 不能给 A 加锁。

②共享锁（Share Locks，简称 S 锁），又称读锁。事务 T 对数据对象 A 加上 S 锁，其它事务只能再对 A 加 S 锁，直到 T 释放 A 上的 S 锁，则事务 T 可以读 A 但不能修改 A。这就保证了其它事务可以读 A，但在 T 释放 A 上的 S 锁之前不能对 A 做任何修改。当 T_1 给 A 加 S 锁时，T_2 只能给个 A 加 S 锁。

（2）时间戳：给每个事务分配一个全局唯一的时间戳。时间戳的值产生精确的顺序，事务按照该顺序提交给 DBMS。时间戳必须保证两个特性：单调性和唯一性。保证时间戳值唯一和单调增长。

（3）乐观控制法：基于数据库操作大部分都不发生冲突假设。乐观法不要求锁定。事务不受限制地被执行直至提交，每个事务经历读、确认、写阶段。

5、什么是封锁引起的活锁问题？如何解决？

活锁问题：可类比操作系统中的饥饿现象。多个事务同时请求封锁数据 R，但始终有事物没有得到系统批准，永远等待中。

避免活锁：采用先来先服务的策略。当多个事务请求封锁同一数据对象时，封锁子系统按请求封锁的先后次序对事务进行排队，数据对象上的锁一旦释放就批准申请队列中第一个事务获得锁。

6、什么是封锁引起的死锁问题？如何解决？

死锁问题：可类比操作系统中的死锁。如果事务 T_1 封锁了数据 R_1，T_2 封锁了数据 R_2，然后 T_1 又请求封锁 R_2，因 T_2 已封锁了 R_2，于是 T_1 等待 T_2 释放 R_2 上的锁。接着 T_2 又请求封锁 R_1，因 T_1 已封锁了 R_1，T_2 也只能等待 T_1 释放 R_1 上的锁。这样就

出现了T_1在等待T_2，而T_2又在等待T_1的局面，T_1和T_2两个事务永远不能结束，形成死锁。

解决方法：

（1）死锁的预防

破坏产生死锁的条件。预防死锁通常有两种方法：

①一次封锁法

一次封锁法要求每个事务必须一次将所有要使用的数据全部加锁，否则就不能继续执行。缺点：扩大了封锁范围，减少系统并发度；第二，数据库中数据是不断变化的，很难事先精确地确定每个事务所要封锁的数据对象，为此需要把可能需要封锁的对象全部封锁，进一步降低了并发度。

②顺序封锁法

所有事务都按预先规定封锁顺序实行封锁。缺点：一，数据库中封锁的数据对象极多，且不断变化，维护困难，成本高。二，事务的封锁请求动态变化，难以事先确定封锁对象，因此难以按照事先规定的顺序进行封锁。可见，在操作系统中广为采用的预防死锁的策略并不很适合数据库的特点，因此 DBMS 普遍采用的是诊断并解除死锁的方法。

（2）死锁的诊断与解除

数据库系统中诊断死锁的方法与操作系统类似，一般使用两种方法：

①超时法

预先设定一个时限，如果事务的等待时间超过了规定的时限，就认为发生了死锁。超时法实现简单，缺点：有可能误判死锁，因为超时可能是由于别的原因导致；时限难以确定，若时限设置太长，可能有些死锁不能及时发现。

②等待图法

类比操作系统中资源分配图。有环则有死锁现象。

系统在检测到死锁后，解除死锁方法通常是撤销一个处理死锁代价最小的事务，并释放此事务持有的所有的锁，使其它事务得以继续运行下去。同时对撤销的事务所执行的数据修改操作必须加以恢复。

7、关系模型的形式化定义是什么？什么是关系？

关系模型只包含单一的数据结构——关系。关系是二维表的超集。下面从集合论角度给出关系数据结构的形式化定义。

（1）域（Domain）

域是一组具有相同数据类型的值的集合。

笛卡尔积（Cartesian Product）

笛卡尔积是域上的一种集合运算。

给定一组域$D_1,D_2,\cdots D_n$，这些域中可以存在相同的域。$D_1,D_2,\cdots D_n$的笛卡尔积为$D_1\times D_2\times\cdots\times D_n=\{(d_1,d_2,\cdots d_n)\mid d_i\in D_i,i=1,2,\cdots,n\}$，其中每一个元素$(d_1,d_2,\cdots d_n)$叫做一个$n$元组（n-tuple）或简称元组（Tuple）。元素中的每一个值d_i叫做一个分量（Component）。

若$D_i(i=1,2,\cdots,n)$为有限集，其基数（Cardinal number）为$m_i(i=1,2,\cdots,n)$，

则$D_1\times D_2\times\cdots\times D_n$的基数$M$为$M=\prod_{i=1}^{n}m_i$。笛卡尔积可表示为一个二维表。表中的每行对应一个元组，表中的每一列的值来自一个域。

（3）关系（Relation）

$D_1\times D_2\times\cdots\times D_n$的子集叫做在域$D_1,D_2,\cdots D_n$上的关系，表示为$R(D_1,D_2,\cdots D_n)$。这里$R$表示关系的名字，$n$是关系的目或度（Degree）。关系中的每个元素是关系中的元组，通常用t表示。

当$n=1$时，称该关系为单元关系（Unary relation），或一元关系。

当$n=2$时，称该关系为二元关系（Binary relation）。

关系是笛卡尔积的有限子集，所以关系也是一个二维表，表的每行对应一个元组，表的每一列对应一个域。由于域可以相同，为了加以区分，必须对每列起一个名字，称为属性（Attribute）。n目关系必有n个属性。

关系可以有三种类型：基本关系（通常又称为基本表或基表）、查询表和视图表。基本表是实际存在的表，它是实际存储数据的逻辑表示。查询表是查询结果对应的表。视图表是由基本表或其它视图表导出的表，是虚表，不对应实际存储的数据。

基本关系具有以下 6 条性质：

①列是同质（Homogeneous）的，即每一列中的分量是同一类型的数据，来自同一个域。

②不同的列可出自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。

③列的顺序无所谓，即列的次序可以任意交换。在许多实际关系数据库产品中，增加新属性时，永远是插至最后一列。

④任意两个元组的候选码不能相同。

⑤行的顺序无所谓，即行的次序可以任意交换。

⑥分量必须取原子值，即每一个分量都必须是不可分的数据项。

关系模型要求关系必须规范化（normalization）的，即要求关系必须满足一定的规范条件。其中最基本的一条是，关系的每一个分量必须是一个不可分的数据项。规范化的关系简称为范式（Normal Form）。

8、什么是数据库镜像？

数据库镜像是 DBMS 根据 DBA 的要求，自动把整个数据库或其中的关键数据复制到另一个磁盘上，每当主数据库更新时，DBMS 会自动把更新后的数据复制过去，即 DBMS 自动保证镜像数据与主数据的一致性。

作用：

①为避免磁盘介质故障影响数据库的可用性。一旦出现介质故障，可由镜像磁盘继续提供使用，同时 DBMS 自动利用镜像磁盘数据进行数据库的恢复，不需要关闭系统和重装数据库副本。

②在没有出现故障时，数据库镜像可用于并发操作，即当一个用户对数据加排它锁修改数据时，其它用户可以读镜像数据库上的数据，而不必等待该用户释放锁。

由于数据库镜像是通过复制数据实现的，频繁地复制数据自然会降低系统运行效率，因此在实际应用中用户往往只选择对关键数据和日志文件镜像，而不是

对整个数据库进行镜像。

9、什么是关系代数？它有哪些运算？

关系代数是一种抽象的查询语言，它用对关系的运算来表达查询。运算对象、运算符、运算结果是运算的三大要素。关系代数的运算对象是关系，运算结果亦为关系。关系代数用到的运算符包括四类，集合运算符、专门的关系运算符、算术比较符和逻辑运算符。

关系代数的运算按运算符的不同可分为传统的集合运算和专门的关系运算两类。其中传统的集合运算将关系看成元组的集合，其运算是从关系的“水平”方向即行的角度来进行。而专门的关系运算不仅涉及行而且涉及列。

（1）传统的集合运算

并、差、交、笛卡尔积四种运算。

（2）专门的关系运算

专门的关系运算包括选择、投影、连接、除运算等。

10、非关系型数据库有哪些特点？

非关系型数据库的产生主要是为了解决大数据的应用问题。

对于非关系型数据库并没有一个明确的范围和定义，但是它们都普遍存在如下特点：

（1）不需要预定义模式

数据中的每条记录都可能有不同的属性和格式。插入数据时，不需要事先定义数据模式和表结构。

（2）无共享架构

相对于将所有数据存储的存储区域网络中的全共享架构，非关系型数据库往往将数据划分后存储在各个本地服务器上。因为从本地磁盘读取数据的性能往往好于通过网络传输读取数据的性能，从而提高了系统的性能。

（3）弹性可扩展

可以在系统运行的时候，动态增加或者删除结点。不需要停机维护，数据可以自动迁移。

（4）需要分区

相对于将数据存放于同一个节点，非关系型数据库需要将数据进行分区，将记录分散在多个节点上面，并且通常分区的同时还要做复制。这样既提高了并行性能，又能保证没有单点失效的问题。

（5）异步复制

和 RAID 存储系统不同的是，非关系型数据库中的复制，往往是基于日志的异步复制。这样数据就可以尽快地写入一个节点，而不会被网络传输引起迟延。缺点是并不总是能保证一致性，这样的方式在出现故障的时候，可能会丢失少量的数据。

（6）BASE 模型

相对于事务严格的 ACID 特性，非关系型数据库保证的是 BASE 特性，其含义如下：

“BA”即基本可用（Basically Available），支持分区失败，如 sharding 碎片划分数据库；“S”即软状态（Soft state），状态可以有一段时间异步；“E”即最终一致（Eventually consistent），最终数据是一致的就可以了，而不是时时一致。

非关系型数据库适用情况：数据模型比较简单；需要灵活性更强的 IT 系统；

对数据库性能要求较高；不需要高度的数据一致性；对于给定关键码，比较容易映射复杂值的环境。

3.3.8 编译原理

3.3.9 通信原理

3.3.10 其他特殊面试问题集锦