1. 以下叙述中错误的是(　　)。

A) 可以通过typedef增加新的类型

B) 可以用typedef将已存在的类型用一个新的名字来代表

C) 用typedef定义新的类型名后，原有类型名仍有效

D) 用typedef可以为各种类型起别名，但不能为变量起别名

参考答案：A

【解析】关键字typedef的作用只是将C语言中的已有的数据类型作了置换，并不是增加新的类型，所以A)错误。

2．以下关于typedef的叙述错误的是

A) 用typedef可以增加新类型

B) typedef只是将已存在的类型用一个新的名字来代表

C) 用typedef可以为各种类型说明一个新名，但不能用来为变量说明一个新名

D) 用typedef为类型说明一个新名，通常可以增加程序的可读性

参考答案：A

【解析】typedef并不是增加了新类型，而是用一个新名字替代已存在的类型，不能为变量说明一个新名，使用typedef可以增强移植性。所以A选项错误。

3. 若有以下语句

typedef struct S

{ int g; char h; } T;

以下叙述中正确的是

A) 可用S定义结构体变量 B) 可用T定义结构体变量

C) S是struct 类型的变量 D) T是struct S类型的变量

参考答案：B

【解析】本题考查typedef重新声明一种结构体类型，那么T为结构体类型，而不是结构体变量，所以B选项正确。

4. 设有以下语句

typedef struct TT

{ char c; int a[4]; } CIN;

则下面叙述中正确的是

A) CIN是struct TT类型的变量 B) TT是struct类型的变量

C) 可以用TT定义结构体变量 D) 可以用CIN定义结构体变量

参考答案：D

【解析】本题考查typedef重新声明一种结构体类型，其中CIN为结构体类型名，而不是结构体变量，所以D选项正确。

5. 以下叙述中错误的是

A) 可以用typedef将已存在的类型用一个新的名字来代表

B) 可以通过typedef增加新的类型

C) 用typedef定义新的类型名后，原有类型名仍有效

D) 用typedef可以为各种类型起别名，但不能为变量起别名

参考答案：B

【解析】本题考查typedef的用法，typedef并不是增加一种新的类型，而是对已存在的类型用一个新的名字来代表，所以B选项错误。

6. 若有定义

typedef int\* T;

T a[10];

则a的定义与下面哪个语句等价

A) int (\*a)[10]; B) int \*a[10]; C) int \*a; D) int a[10];

参考答案：B

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，其中本题中int\* 用T来代替，所以定义T a[10];就相当于是int \*a[10];选项B正确。

7. 若有定义

typedef char T[10];

T \*a;

上述定义中a的类型与下面选项中完全相同的是

A) char a[10]; B) char (\*a)[10]; C) char \*a; D) char \*a[10];

参考答案：D

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，选项D正确。

8. 以下结构体说明和变量定义中,正确的是

A) typedef struct abc{ int n;double m; }ABC; B) struct abc{ int n;double m };

ABC x,y; struct abc x, y;

C) struct ABC{ int n;double m; } D) struct abc{ int n;double m; };

struct ABC x, y; abc x, y;

参考答案：A

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，选项A中ABC为新的类型别名，定义正确。

9. 以下叙述中正确的是

A) 使用typedef说明新类型名时，其格式是： typedef 新类型名 原类型名;

B) 在程序中，允许用typedef来说明一种新的类型名

C) 使用typedef 说明新类型名时，后面不能加分号

D) 在使用typedef改变原类型的名称后，只能使用新的类型名

参考答案：B

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，选项A总新类型名应该在原类型名之后，C选项中后面要加分号，D选项中可以使用原类型名。

10. 以下叙述中正确的是

A) 使用typedef定义新类型名后，新类型名与原类型名实际上是等价的

B) 结构体类型中的各个成分均不能是数组或指针

C) 结构体类型的变量，不能在声明结构体类型组成时一起定义

D) 元素为结构体类型的数组，只能在声明过结构体类型之后，单独进行定义

参考答案：A

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，结构体类型中的成分可以是数组和指针，所以B选项错误，结构体类型的变量可以在声明结构体的时候一起定义，C选项错误，D选项中可以一起定义。

11. 以下叙述中错误的是

A) 用typedef可以说明一种新的类型名

B) typedef的作用是用一个新的标识符来代表已存在的类型名

C) 可以用typedef说明的新类型名来定义变量

D) typedef说明的新类型名必须使用大写字母，否则会出编译错误

参考答案：D

【解析】本题考查typedef的使用方法，typedef对已存在的类型使用一个新的名字，新类型可以使用小写，所以D选项错误。

12. 有以下程序

#include <stdio.h>

main()

{ struct STU { char name[9]; char sex; double score[2]; };

struct STU a={"Zhao", 'm', 85.0, 90.0}, b={"Qian", 'f', 95.0, 92.0};

b=a;

printf("%s,%c,%2.0f,%2.0f\n", b.name, b.sex, b.score[0], b.score[1]);

}

程序的运行结果是

A) Qian,m,85,90 B) Zhao,m,85,90 C) Zhao,f,95,92 D) Qian,f,95,92

参考答案：B

【解析】本题考查结构体初始化操作，本题中可以直接将结构体a赋值给b，所以输出的结果和a一样，选项B正确。

13. 以下关于C语言数据类型使用的叙述中错误的是

A) 若要处理如"人员信息"等含有不同类型的相关数据，应自定义结构体类型

B) 若要保存带有多位小数的数据，可使用双精度类型

C) 若只处理"真"和"假"两种逻辑值，应使用逻辑类型

D) 整数类型表示的自然数是准确无误差的

参考答案：C

【解析】C语言中没有逻辑类型,所以C错误｡若要保存带有多位小数的数据,可以用单精度类型也可以用双精度类型｡处理包含不同类型的相关数据可以定义为结构体类型｡整数类型可以无误差的表示自然数｡

14. 以下结构体类型说明和变量定义中正确的是

A) struct REC ; B) typedef struct

{ int n; char c; }; { int n; char c; } REC;

REC t1,t2; REC t1,t2;

C) typedef struct REC; D) struct

{ int n=0; char c='A'; } t1,t2; { int n; char c; } REC;

REC t1,t2;

参考答案：B

【解析】本题考查结构体的相关知识，选项A中struct REC后面不能有分号，C选项中typedef struct REC的后面也不能有分号，选项D中REC已经是结构体变量，不能当做结构体类型来使用。

15. 以下叙述中正确的是

A) 结构体类型中各个成分的类型必须是一致的

B) 结构体类型中的成分只能是C语言中预先定义的基本数据类型

C) 在定义结构体类型时，编译程序就为它分配了内存空间

D) 一个结构体类型可以由多个称为成员（或域) 的成分组成

参考答案：D

【解析】本题考查结构体类型，结构体类型中的成分可以是结构体，所以B选项错误，定义结构体时编译程序并不会给它分配内存空间，所以C选项错误，结构体中各个成分的类型可以是不一样的，所以A选项错误。

16. 下面结构体的定义语句中，错误的是(　　)。

A) struct ord {int x；int y；int z；} struct ord a；

B) struct ord {int x；int y；int z；}； struct ord a；

C) struct ord {int x；int y；int z；} a；

D) struct {int x；int y；int z；} a；

参考答案：A

【解析】A)选项struct ord {int x；int y；int z；} struct ord a；错误，不能在定义结构体的同时，又用结构体类型名定义变量。应该写成B)选项或者D)选项的格式。

17．设有定义：

struct complex

{　int real，unreal；} data1＝{1,8}，data2；

则以下赋值语句中错误的是(　　)。

A) data2＝(2,6)； B) data2＝data1；

C) data2.real＝data1.real； D) data2.real＝data1.unreal；

参考答案：A

【解析】A)选项中可以在声明变量的同事为data2赋值，但是 data2＝(2,6)；应写作data2＝{2,6}。所以选择A)。

18．有以下程序

#include <stdio.h>

#include <string.h>

struct A

{ int a;

char b[10];

double c;

};

struct A f(struct A t);

main()

{ struct A a={1001,"ZhangDa",1098.0};

a=f(a);

printf("%d,%s,%6.1f\n",a.a,a.b,a.c);

}

struct A f(struct A t)

{ t.a=1002;

strcpy(t.b,"ChangRong");

t.c=1202.0;

return t;

}

程序运行后的输出结果是

A) 1002,ZhangDa,1202.0 B) 1002,ChangRong,1202.0

C) 1001,ChangRong,1098.0 D) 1001,ZhangDa,1098.0

参考答案：B

【解析】本题考查结构体变量的引用以及作为函数参数，题目虽然看似复杂，其实比较容易，f函数的参数是结构体变量，然后对参数重新赋值并返回，所以该题目的答案为B选项。

19.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->x B) pt->x C) \*pt->y D) (pt++)->x

参考答案：A

【解析】本题考查结构体数组初始化以及结构体变量的引用，题目中定义了一个结构体数组c并初始化，指针pt指向c的第一个元素，那么pt->x为10，++pt->x为11，\*pt->y为1，(pt++)->x为20，所以答案为A。

20. 有以下程序

#include <stdio.h>

struct S{ int n; int a[20]; };

void f(int \*a, int n)

{ int i;

for (i=0; i<n-1; i++)

a[i]+=i;

}

main()

{ int i;

struct S s={10, {2,3,1,6,8,7,5,4,10,9}};

f(s.a, s.n);

for (i=0; i<s.n; i++)

printf("%d,", s.a[i]);

}

程序运行后的输出结果是

A) 2,3,1,6,8,7,5,4,10,9, B) 3,4,2,7,9,8,6,5,11,10,

C) 2,4,3,9,12,12,11,11,18,9, D) 1,2,3,6,8,7,5,4,10,9,

参考答案：C

【解析】题目中定义了一个结构体类型S，然后定义了一个结构体变量s并初始化，执行f函数后，将s中元素a的每个元素都加上i，这里需要主要，最后一个元素没有加i，所以最终答案为C选项。

21. 有以下程序

#include <stdio.h>

#include <string.h>

typedef struct { char name[9]; char sex; float score[2]; } STU;

void f(STU A)

{ STU b={"Zhao", 'm', 85.0, 90.0};

int i;

strcpy(a.name, b.name);

a.sex = b.sex;

for (i=0; i<2; i++)

a.score[i] = b.score[i];

}

main()

{ STU c={"Qian", 'f', 95.0, 92.0};

f(c);

printf("%s,%c,%2.0f,%2.0f\n", c.name, c.sex, c.score[0], c.score[1]);

}

程序的运行结果是

A) Zhao,m,85,90 B) Qian,m,85,90 C) Zhao,f,95,92 D) Qian,f,95,92

参考答案：D

【解析】本题考查结构体的相关操作以及传值、传址的区别，该题中调用f函数后，会生成参数c的一个副本，而不会改变c的值，所以c值维持原值，选项D正确。

22. 有以下程序

#include <stdio.h>

#include <string.h>

struct A

{ int a;

char b[10];

double c;

};

void f(struct A t);

main()

{ struct A a={1001,"ZhangDa",1098.0};

f(a);

printf("%d,%s,%6.1f\n", a.a,a.b,a.c);

}

void f(struct A t)

{ t.a=1002;

strcpy(t.b,"ChangRong");

t.c=1202.0;

}

程序运行后的输出结果是

A) 1002, ZhangDa,1202.0 B) 1002,ChangRong,1202.0

C) 1001, ChangRong,1098.0 D) 1001,ZhangDa,1098.0

参考答案：D

【解析】本题考查结构体的相关操作以及传值、传址的区别，该题中调用f函数后，会生成参数a的一个副本，而不会改变a的值，所以a值维持原值，选项D正确。

23. 有以下定义和语句

struct workers

{ int num; char name[20]; char c;

struct

{ int day; int month; int year;} s;

};

struct workers w,\*pw;

pw=&w;

能给w中year成员赋1980的语句是

A) pw->year=1980; B) w.year=1980; C) w.s.year=1980; D) \*pw.year=1980;

参考答案：C

【解析】本题考查结构体变量的引用，题目中定义了一个结构体，其中结构体中的变量又是一个结构体，w为外层结构体，如果给内层结构体赋值，C先要得到内层结构体变量，即w.s，若要给year赋值，表示为w.s.year即可，选项C正确。

24. 有以下程序

#include <stdio.h>

struct tt

{ int x; struct tt \*y; } \*p;

struct tt a[4]={20,a+1,15,a+2,30,a+3,17,a};

main()

{ int i;

p=a;

for(i=1; i<=2; i++) { printf("%d,", p->x ); p=p->y; }

}

程序的运行结果是

A) 20,30, B) 30,17 C) 15,30, D) 20,15,

参考答案：D

【解析】本题考查结构体变量的引用以及结构体数组，p指向a数组的第一个元素，所以p->x为20，然后p=p->y后，p指向数组a的第二个元素，所以输出15，选项D正确。

25. 设有定义：

struct {char mark[12]; int num1; double num2;} t1,t2;

若变量均已正确赋初值，则以下语句中错误的是

A) t1=t2; B) t2.num1=t1.num1;

C) t2.mark=t1.mark; D) t2.num2=t1.num2;

参考答案：C

【解析】本题考查结构体变量的引用，C选项中mark为结构体中的数组，不能直接赋值，所以C选项错误。

26. 有以下程序

#include <stdio.h>

#include <string.h>

struct A

{ int a; char b[10]; double c;};

void f(struct A t);

main()

{ struct A a={1001,"ZhangDa",1098.0};

f(a); printf("%d,%s,%6.1f\n", a.a,a.b,a.c);

}

void f(struct A t)

{ t.a=1002; strcpy(t.b,"ChangRong"); t.c=1202.0;}

程序运行后的输出结果是

A) 1002,ZhangDa,1202.0 B) 1002,ChangRong,1202.0

C) 1001,ChangRong,1098.0 D) 1001,ZhangDa,1098.0

参考答案：D

【解析】本题考查结构体的相关操作以及传值、传址的区别，该题中调用f函数后，会生成参数a的一个副本，而不会改变a的值，所以a值维持原值，选项D正确。

27. 有以下程序

#include <stdio.h>

#include <string.h>

typedef struct { char name[9]; char sex; float score[2]; } STU;

STU f(STU A)

{ STU b={"Zhao", 'm', 85.0, 90.0}; int i;

strcpy(a.name, b.name);

a.sex = b.sex;

for (i=0; i<2; i++) a.score[i] = b.score[i];

return a;

}

main()

{ STU c={"Qian", 'f', 95.0, 92.0}, d;

d=f(c);

printf("%s,%c,%2.0f,%2.0f\n",d.name,d.sex,d.score[0],d.score[1]);

}

程序的运行结果是

A) Zhao,m,85,90 B) Qian,m,85,90 C) Qian,f,95,92 D) Zhao,f,95,92

参考答案：A

【解析】本题考查结构体的相关操作以及传值、传址的区别，该题中调用f函数后，会返回记过计算的a值，选项A正确。

28. 若有以下程序

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct stu {

char \*name, gender;

int score;

};

main()

{ struct stu a={NULL, 'm', 290}, b;

a.name=(char \*)malloc(10);

strcpy(a.name, "Zhao");

b = a; b.gender = 'f'; b.score = 350;

strcpy(b.name, "Qian");

printf( "%s,%c,%d,", a.name, a.gender, a.score );

printf( "%s,%c,%d\n", b.name, b.gender, b.score );

}

则程序的输出结果是

A) Zhao,m,290,Zhao,f,350 B) Zhao,m,290,Qian,f,350

C) Qian,f,350,Qian,f,350 D) Qian,m,290,Qian,f,350

参考答案：D

【解析】本题中考查结构体中变量的引用，a结构体中name被修改为Qian，所以本题答案为D。

29. 若有以下程序

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct stu {

char \*name, gender;

int score;

} STU;

void f(char \*p)

{ p=(char \*)malloc(10);

strcpy(p, "Qian");

}

main()

{ STU a={NULL, 'm', 290}, b;

a.name=(char \*)malloc(10);

strcpy( a.name, "Zhao" );

b = a;

f(b.name);

b.gender = 'f'; b.score = 350;

printf("%s,%c,%d,", a.name, a.gender, a.score);

printf("%s,%c,%d\n", b.name, b.gender, b.score);

}

则程序的输出结果是

A) Zhao,m,290,Qian,f,350 B) Zhao,m,290,Zhao,f,350

C) Qian,f,350,Qian,f,350 D) Qian,m,290,Qian,f,350

参考答案：B

【解析】本题考查结构体变量中的引用，其中a的name为Zhao，b执行函数f(b.name)后b的name并没有改变，还是Zhao，所以答案为B选项。

30. 若有以下程序

#include <stdio.h>

typedef struct stu {

char name[10], gender;

int score;

} STU;

void f(STU a, STUB)

{ b = a;

printf( "%s,%c,%d,", b.name, b.gender, b.score );

}

main()

{ STU a={"Zhao", 'm', 290}, b={"Qian", 'f', 350};

f(a,b);

printf("%s,%c,%d\n", b.name, b.gender, b.score);

}

则程序的输出结果是

A) Qian,f,350,Qian,f,350 B) Zhao,m,290, Zhao,m,290

C) Zhao,m,290,Qian,f,350 D) Zhao,m,290,Zhao,f,350

参考答案：C

【解析】本题考查结构体的相关操作以及传值、传址的区别，该题中调用f函数后，会首先输出被重新复制的b值，且与a相同，而执行完f函数后，b值并不会改变，所以还是维持原值，选项C正确。