一、结构体类型和结构体变量

1. 单选

- 在定义一个结构体变量时系统分配给它的存储空间是：_____。
  A) 该结构体中第一个成员所需要的存储空间
  B) 该结构体中最后一个成员所需要的存储空间
  C) 该结构体中占用最大存储空间的成员所需要的存储空间
  D) 该结构体中所有成员所需要的存储空间
- 结构体变量在程序执行期间_____。
  A) 所有成员一直驻留在内存中
  B) 只有一个成员驻留在内存中
  C) 部分成员驻留在内存中
  D) 没有成员驻留在内存中
- 基于以下 C 语句，下面叙述中正确的是：_____。

```
typedef struct
{
    int n;
    char ch[8];
}PER;
```

  A) PER 是结构体变量名
  B) PER 是结构体类型名
  C) typedef struct 是结构体类型
  D) struct 是结构体类型名
- 有如下定义：

```
struct Date
{
   int year;
   int month;
   int day;
}
struct worklist
{
   char name[20];
   char sex;
   struct Date birthday;
}person;
```

  对结构体变量 person 的出生年份进行赋值时，下面正确的赋值语句是：_____。
  A) year=1958
  B) birthday.year=1958
  C) person.birthday.year=1958
  D) worklist.Data.year=1958

- 对于以下的变量定义，表达式_____是不正确的。

```
struct node
{
   float x, y;
   char s[10];
}point, *p=&point;
```

A) p->x=2.0
B) (*p).y=3.0
C) point.x=2.0
D) p->s="a"

- 设有如下定义：

```
struct Sk
{
   int a;
   float b;
}data, *p;
```

若有 p=&data，则对 data 中的 a 域的正确引用是：_____。
A) (*p).data.a
B) (*p).a
C) p->data.a
D) p.data.a

- 以下对结构体变量 st 中成员 age 的非法引用是：_____。

```
struct stud
{
   int age;
   int num;
}st, *p=&st;
```

A) st.age
B) stud.age
C) p->age
D) (*p).age

- 对于以下的变量定义，表达式_____在语法和语义上都是正确的。

```
struct node
{
   float x, y;
   char s[10];
}point={1, 2, “abc”}, *p;
```

A) *p=point;
B) p=&point;
C) point=p
D) p->s=point.y;

- 有如下定义：

```
struct Sk
{
```

```
    int a;
    float b;
}data;
int *p;
```

若要使 p 指向 data 中的 a 成员，正确的赋值语句是：_____。

A) p=&a;

B) p=data.a;

C) p=&data.a;

D) *p=data.a;

2. 判断

- 结构体成员的类型必须是基本数据类型。
- 结构体类型数据由多个成员构成，这些成员的类型可以不同，它们共同描述一个对象。
- 在程序中定义了一个结构体类型后，可以多次用它来定义具有该类型的变量。
- C 语言中，结构体类型和结构体变量的含义一样，都可以用来存放数据。
- C 语言中，结构体的成员可以是一维数组或多维数组。
- 使几个不同的变量共占同一段内存的结构，称为结构体类型。
- 对于不同类型的数据，若想合成一个有机的整体，可以引用结构体进行定义。
- 结构体变量占用的内存空间为所有成员占用的空间之和。
- 在引用结构体成员时，只能对最低级的成员进行赋值或存取操作或计算。
- 不同结构体（类型）的成员必须有唯一名称。
- 不能使用运算符==和!=来比较同一结构体类型的两个结构体变量。

3. 程序运行结果

- 分析以下程序的输出结果。

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
struct STUD
{
    int no;
    char *name;
    int score;
};
int main()
{
    struct STUD st1={1, "Mary", 85}, st2;
    st2.no=2;
    st2.name=(char *)malloc(sizeof(10));
    strcpy(st2.name, "Smith");
    st2.score=78;
```

```
    printf("%s\n", (st1.score>st2.score?st1.name:st2.name));
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct STUD
{
    int no;
    struct STUD *next;
};
int main()
{
    int i;
    struct STUD st1, st2, st3, *st;
    st1.no=1; st1.next=&st2;
    st2.no=2; st2.next=&st3;
    st3.no=3; st3.next=&st1;
    st=&st1;
    for(i=1; i<=4; i++)
    {
        printf("%d", st->no);
        st=st->next;
    }
    printf("\n");
    return 0;
}
```

4. 程序完形填空

- 若有以下的结构体类型声明：

```
struct STRU
{
    int a, b;
    char c;
    double d;
    struct STRU *p1, *p2;
};
```

_____

请填空，以完成对 t 数组的定义，t 数组的每个元素为该结构体类型，且有 20 个元素。

二、结构体数组和结构体指针

1. 单选

- 有如下定义：

```
struct person
{
    char name[9];
    int age;
};
struct person class[10]={"Johu", 17, "Paul", 19, "Mary", 18, "Adam", 16};
```

- 根据上述定义，能输出字母 M 的语句是：_____。

A) printf("%c\n", class[3].name);
B) printf("%c\n", class[3].name[1]);
C) printf("%c\n", class[2].name[1]);
D) printf("%c\n", class[2].name[0]);

- 设结构体变量的定义如下，则对其中的结构体成员 num 的正确引用是：_____。

```
struct Student
{
    int num;
    char name[20];
    float score;
}stud[10];
```

A) stud[1].num=10;
B) Student.stud.num=10;
C) struct.stud.num=10;
D) struct.Student.num=10;

- 若有以下 C 语句，则表达式_____中的值为 101。

```
struct we
{
    int a;
    int *b;
}*p;
int x0[]={11, 12}, x1[]={31, 32};
struct we x[2]={100, x0, 300, x1};
p=x;
```

A) *p->b
B) p->a
C) ++p->a
D) (p++)->a

- 若有以下 C 语句，则表达式_____中的值为 300。

```
struct we
{
```

```
    int a;
    int *b;
}*p;
int x0[]={11, 12}, x1[]={31, 32};
struct we x[2]={100, x0, 300, x1};
p=x;
```

A) ++(p->a)

B) (++p)->a

C) ++p->a

D) (p++)->a

- 若有以下 C 语句，则表达式_____中的值为 100。

```
struct we
{
    int a;
    int *b;
}*p;
int x0[]={11, 12}, x1[]={31, 32};
struct we x[2]={100, x0, 300, x1};
p=x;
```

A) ++(p->a)

B) (++p)->a

C) ++p->a

D) (p++)->a

- 设有以下语句：

```
struct st
{
   int n;
   struct st *next;
};
static struct st a[3]={5, &a[1], 7, &a[2], 9, '\0'}, *p;
p=&a[0];
```

则表达式_____的值是 6。

A) p++->n

B) p->n++

C) (*p).n++

D) ++p->n

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
struct st
{
   int x;
   int *y;
}*p;
int dt[4]={10, 20, 30, 40};
```

```
struct st aa[4]={50, &dt[0], 60, &dt[1], 70, &dt[2], 80, &dt[3]};
int main()
{
   p=aa;
   printf("%d,", ++p->x);
   printf("%d,", (++p)->x);
   printf("%d\n", ++(*p->y));
   return 0;
}
```

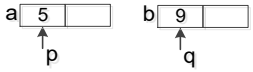
A) 10, 20, 20

B) 50, 60, 21

C) 51, 60, 21

D) 60, 70, 31

- 有以下如图所示的结构体声明和结构体变量定义，指针 p 指向变量 a，指针 q 指向变量 b。不能把结点 b 连接到结点 a 之后的语句是：_____。




```
struct node
{
   char data;
   struct node *next;
}a, b, *p=&a, *q=&b;
```

A) a.next=q;

B) p.next=&b;

C) p->next=&b;

D) (*p).next=q;

- 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));

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
struct HAR
{
   int x, y;
   struct HAR *p;
```

```
}h[2];
int main()
{
   h[0].x=1; h[0].y=2;
   h[1].x=3; h[1].y=4;
   h[0].p=&h[1];
   h[1].p=h;
   printf("%d %d\n", (h[0].p)->x, (h[1].p)->y);
   return 0;
}
```

A) 1 2
B) 2 3
C) 1 4
D) 3 2

2. 判断

- 结构体变量可以作数组元素。
- 可以把结构体数组元素作为一个整体输出。
- 结构体数组不可以在定义时进行初始化。
- 结构体数组中可以包含不同结构体类型的结构体变量。
- 必须先申请结构体类型，才能定义相应的结构体数组。
- 结构体指针可以存放任一结构体变量的地址。

3. 填空

- 若有以下的结构体类型声明与结构体变量定义，可用 a.day 引用结构体成员 day，请写出引用结构体成员 a.day 的其他两种形式_____和_____。

```
struct
{
   int day;
   char mouth;
   int year;
}a, *b;
b=&a;
```

- 有如下定义：

```
struct
{
   int x;
   int *y;
}tab[2]={{1, "ab"}, {2, "cd"}}, *p=tab;
```

则表达式*p->y 的结果是：（1）；表达式*(++p)->y 的结果是：（2）。

- 有如下定义：

```
struct
```

```
{
    int x;
    int y;
}s[2]={{1,2}, {3,4}}, *p=s;
```

则表达式++p->x 的结果是：（1）；表达式(++p)->x 的结果是：（2）。

4. 简答

- 说明为什么以下定义是错误的：

```
struct Node1
{
    int data;
    struct Node1 next;
};
```

而以下定义是正确的：

```
struct Node2
{
    int data;
    struct Node2 *next;
};
```

- 有如下 C 语言变量定义：

```
struct
{
    int i, j;
}a={0, -1}, *p=&a;
unsigned k;
double x;
```

- 请分别给出以下各表达式的值：
  - （1）a.i++?a.i--:a.i++
  - （2）k=a.i--+a.j--
  - （3）p->i && ~p->j
  - （4）x=--p->j>>1
  - （5）++(*p).i, !(*p).j

5. 程序运行结果

- 以下程序的执行结果是：_____。

```
#include <stdio.h>
struct s
{
    int n;
    int *m;
}*p;
int d[5]={30,10,40,20,50};
```

```
struct s arr[5]={300, &d[0], 100, &d[1], 400, &d[2], 200, &d[3], 500, &d[4]};
int main()
{
    p=arr;
    printf("%d,", ++p->n);
    printf("%d,", (++p)->n);
    printf("%d\n", ++(*p->m));
    return 0;
}
```

- 以下程序的执行结果是：_____。

```
#include <stdio.h>
struct st
{
    int x;
    int *y;
}*p;
int s[]={10,20,30,40};
struct st a[]={1,&s[0],2,&s[1],3,&s[2],4,&s[3]};
int main()
{
    p=a;
    printf("%d,", p->x);
    printf("%d,", (++p)->x);
    printf("%d,", *(++p)->y);
    printf("%d\n", ++(*(++p)->y));
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct s
{
    int x;
    int *y;
};
int data[5]={10,20,30,40,50};
struct s a[5]={100,&data[0],200,&data[1],300,&data[2],400,&data[3],500,&data[4]};
int main()
{
    int i=0;
    struct s s_var;
    s_var=a[0];
    printf("%d, ", s_var.x);
    printf("%d, ", *s_var.y);
    printf("%d, ", a[i].x);
```

```
    printf("%d, ", *a[i].y);
    printf("%d, ", ++a[i].x);
    printf("%d, ", ++*a[i].y);
    printf("%d, ", a[++i].x);
    printf("%d, ", *++a[i].y);
    printf("%d, ", (*a[i].y)++);
    printf("%d, ", *(a[i].y++));
    printf("%d, ", *a[i].y++);
    printf("%d\n", *a[i].y);
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct s
{
    int x;
    int *y;
}*p;
int data[5]={10,20,30,40,50};
struct s a[5]={100,&data[0],200,&data[1],300,&data[2],400,&data[3],500,&data[4]};
int main()
{
    p=a;
    printf("%d,", p->x);
    printf("%d,", (*p).x);
    printf("%d,", *p->y);
    printf("%d,", *(*p).y);
    printf("%d,", ++p->x);
    printf("%d,", (++p)->x);
    printf("%d,", p->x++);
    printf("%d,", p->x);
    printf("%d,", ++(*p->y));
    printf("%d,", ++*p->y);
    printf("%d,", *++p->y);
    printf("%d,", p->x);
    printf("%d,", *(++p)->y);
    printf("%d,", p->x);
    printf("%d,", *p->y++);
    printf("%d,", p->x);
    printf("%d,", *(p->y)++);
    printf("%d,", p->x);
    printf("%d,", *p++->y);
    printf("%d\n", p->x);
return 0;
```

```
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct tree
{
    int x;
    char *y;
    struct tree *tpi;
}t[]={{1,"pascal",0},{3,"basic",0}};
int main()
{
    struct tree *p=t;
    char c,*s;
    s=++p->y;
    printf("%s,",s);
    s=++p->y;
    printf("%s,",s);
    c=*p->y;
    printf("%c,",c);
    c=*p++->y;
    printf("%c,",c);
    c=*p->y++;
    printf("%c,",c);
    c=*p->y;
    printf("%c\n",c);
    return 0;
}
```

- 以下程序的输出结果是：

```
#include <stdio.h>
struct S
{
   int a, b;
}data[2]={10,100,20,200};
int main()
{
   struct S p=data[1];
   printf("%d\n", ++(p.a));
   return 0;
}
```

三、函数之间结构体变量的数据传递

1. 单选

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
struct stu
{
    int num;
    char name[10];
    int age;
};
void fun(struct stu *p)
{
    printf("%s\n", (*p).name);
}
int main()
{
    struct stu students[3]={{9801, "Zhang", 20},{9802, "Wang", 19},{9803, "Zhao", 18}};
    fun(students+2);
    return 0;
}
```

A) Zhang
B) Zhao
C) Wang
D) 18

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
struct stu
{
   char name[10];
   int num;
};
void f1(struct stu c)
{
   struct stu b={"LiSiGuo", 2042};
   c=b;
}
void f2(struct stu *c)
{
   struct stu b={"SunDan", 2044};
   *c=b;
}
```

```
int main()
{
    struct stu a={"YangSan", 2041}, b={"WangYin", 2043};
    f1(a);
    f2(&b);
    printf("%d %d\n", a.num, b.num);
    return 0;
}
```

A) 2041 2044
B) 2041 2043
C) 2042 2044
D) 2042 2043

2. 判断

- 结构体变量作为参数时是地址传递。
- 指向结构体变量的指针可以作为函数参数，实现传址调用。

3. 简答

- 以下程序用于输入两个学生的学号和姓名，然后输出。指出并改正程序中的错误。

```
#include <stdio.h>
struct Stud
{
    int no;
    char name[10];
};
void disp(struct Stud s)
{
    printf("%s(%d) ", s.name, s.no);
}
void input(struct Stud s)
{
    printf("学号：");
    scanf("%d", &s.no);
    printf("姓名：");
    scanf("%s", s.name);
}
int main()
{
    struct Stud s[2];
    int i;
    for(i=0; i<2; i++) input(s[i]);
    for(i=0; i<2; i++) disp(s[i]);
```

```
    printf("\n");
    return 0;
}
```

4. 程序运行结果

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct Stud
{
    int no;
    char name[10];
    struct Stud *next;
};
void fun(struct Stud *s)
{
    s=s->next;
}
int main()
{
    int i;
    struct Stud s[2]={{1, "Mary"},{2, "Smith"}};
    struct Stud *h;
    s[0].next=&s[1];
    s[1].next=&s[0];
    h=&s[0];
    fun(h);
    for(i=0; i<2; i++)
    {
        printf("%s(%d) ", h->name, h->no);
        h=h->next;
    }
    printf("\n");
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct Stud
{
    int no;
    char name[10];
    struct Stud *next;
};
void fun(struct Stud **s)
```

```
{
    *s=(*s)->next;
}
int main()
{
    int i;
    struct Stud s[2]={{1, "Mary"},{2, "Smith"}};
    struct Stud *h;
    s[0].next=&s[1];
    s[1].next=&s[0];
    h=&s[0];
    fun(&h);
    for(i=0; i<2; i++)
    {
        printf("%s(%d) ", h->name, h->no);
        h=h->next;
    }
    printf("\n");
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct Stud
{
  int no;
  char name[10];
  struct Stud *next;
};
void fun(struct Stud *s)
{
  int i;
  for(i=0; i<2; i++)
  {
    printf("%s(%d) ", s->name, s->no);
    s=s->next;
  }
  printf("\n");
}
int main()
{
  struct Stud s[2]={{1, "Mary"},{2, "Smith"}};
  struct Stud *h;
  s[0].next=&s[1];
  s[1].next=&s[0];
```

```
    h=&s[0];
    fun(h);
    return 0;
}
```

5. 程序完形填空

- 以下程序的功能是先输入 20 个人的姓名和他们的电话号码（7 位数字），然后输入某人姓名，查找该人的电话号码，请填空：

```
#include <stdio.h>
#include <string.h>
#define N 3
struct ph
{
    char name[20];
    char tel[13];
};
int main()
{
    __（1）__ s[N];
    void readin(struct ph *p);
    void search(struct ph *p, char *x);
    char c[10];
    readin(s);
    printf("请输入被查人的姓名：");
    gets( __（2）__ );
    search(s, c);
    return 0;
}
void readin(struct ph *p)
{
    int i;
    for(i=0; i<N; i++, p++)
    {
        printf("请输入姓名：");
        gets( __（3）__ );
        printf("请输入他的电话号码：");
        gets( __（4）__ );
    }
}
void search(struct ph *p, char *x)
{
    int i;
    struct ph *t=p;
```

```
    for(i=0;i<N;i++,t++)
    {
        printf("%s 的电话号码是：%s\n", t->name, t->tel);
    }
    for(i=0; i<N; i++, p++)
    {
        if(strcmp( （5） )==0)
        {
            printf("%s 的电话号码是：%s\n", x, p->tel);
            break;
        }
    }
    if(i==N) printf("找不到%s 的电话号码\n", x);
}
```

四、链表

1. 单选

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
#include <stdlib.h>
void fun(int **s, int p[2][3])
{
   **s=p[1][1];
}
int main()
{
   int a[2][3]={1,3,5,7,9,11}, *p;
   p=(int *)malloc(sizeof(int));
   fun(&p, a);
   printf("%d\n", *p);
   return 0;
}
```

A) 1
B) 7
C) 9
D) 11

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
#include <stdlib.h>
int a[3][3]={1,2,3,4,5,6,7,8,9}, *p;
void fun(int *s, int b[][3])
```

```
{
   *s=b[1][1];
}
int main()
{
   p=(int *)malloc(sizeof(int));
   fun(p, a);
   printf("%d\n", *p);
   return 0;
}
```

A) 1
B) 4
C) 7
D) 5

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
#include <stdlib.h>
void f(int *s, int b[][3])
{
    *s=b[1][1];
}
int main()
{
    int a[3][3]={1,2,3,4,5,6,7,8,9}, *p;
    p=(int *)malloc(sizeof(int));
    f(p, a);
    printf("%d\n", *p);
    return 0;
}
```

A) 1
B) 4
C) 7
D) 5

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
#include <malloc.h>[1]
void amovep(int *p, int (*a)[3], int n)
{
   int i, j;
   for(i=0; i<n; i++)
      for(j=0; j<n; j++)
      {
         *p=a[i][j];
```

[1] 与使用#include <stdlib.h>效果一样。

```
            p++;
        }
    }
    int main()
    {
        int *p, a[3][3]={{1,3,5}, {2,4,6}};
        p=(int *)malloc(100);
        amovep(p, a, 3);
        printf("%d %d\n", p[2], p[5]);
        free(p);
        return 0;
    }
```

A) 5 6

B) 2 5

C) 3 4

D) 程序错误

- 以下程序运行时，如果从键盘输入 abc def↙，则输出结果是：_____。

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char *p, *q;
    p=(char *)malloc(sizeof(char)*20);
    q=p;
    scanf("%s %s", p, q);
    printf("%s %s\n", p, q);
    return 0;
}
```

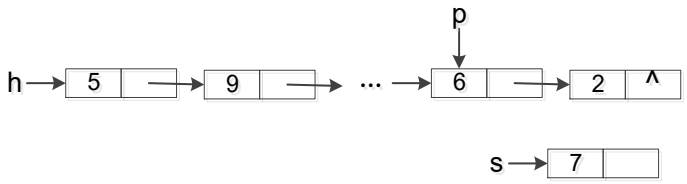
A) def def

B) abc def

C) abc d

D) d d

- 若已建立如图所示的单链表结构，在该链表结构中，指针 p、s 分别指向图中所示结点，现在想把 s 所指的结点插入到链表末尾并且仍然仍构成单链表，则下面不成功的语句是：_____。



A) p=p->next; s->next=p; p->next=s;

B) p=p->next; s->next=p->next; p->next=s;

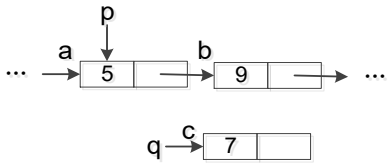
C) s->next=NULL; p=p->next; p->next=s;

D) p=(*p).next; (*s).next=(*p).next; (*p).next=s;

- 有以下定义：

```
struct link
{
   int data;
   struct link *next;
}a, b, c, *p, *q;
```

且变量 a 和 b 之间已有如图所示的链表结构。指针 p 指向变量 a，指针 q 指向变量 c。则无法完成把 c 插入到 a 和 b 之间以形成新链表的语句是：_____。



A) a.next=c; c.next=b;

B) q->next=p->next; p->next=q;

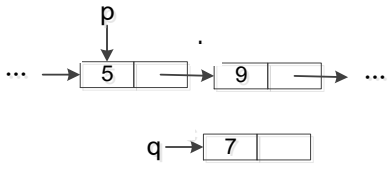
C) p->next=&c; q->next=&b;

D) (*p).next=q; (*q).next=&b;

- 有以下定义：

```
struct link
{
   int data;
   struct link *next;
}*p, *q;
```

且指针 p、q 指向的点之间已有如图所示的链表结构。实现将 q 指向点插入 p 指向点后面、并形成新链表的语句是：_____。



A) p=p->next; p->next=q;

B) q->next=p->next; p->next=q;

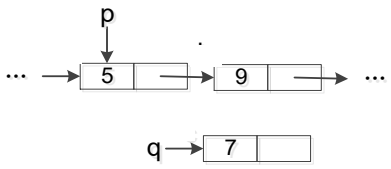
C) p->next=q; q->next=p->next;

D) q->next=p->next->next; p->next=q;

- 有以下定义：

```
struct link
{
   int data;
   struct link *next;
}*p, *q;
```

且指针 p、q 指向的点之间已有如图所示的链表结构。实现将 q 指向点取代 p 指向点的“后续点”、并形成新链表的语句是：_____。



A) p=p->next; p->next=q;
B) q->next=p->next; p->next=q;
C) p->next=q; q->next=p->next;
D) q->next=p->next->next; p->next=q;

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
#include <malloc.h>
struct NODE
{
   int num;
   struct NODE *next;
};
int main()
{
   struct NODE *p, *q, *r;
   p=(struct NODE *)malloc(sizeof(struct NODE));
   q=(struct NODE *)malloc(sizeof(struct NODE));
   r=(struct NODE *)malloc(sizeof(struct NODE));
   p->num=10; q->num=20; r->num=30;
   p->next=q; q->next=r; r->next=NULL;
   printf("%d\n", p->num+q->next->num);
   return 0;
}
```

A) 10
B) 20
C) 30
D) 40

2. 判断

- 有如下 C 语句，判断说法的正误：内存动态分配语句执行后将不能使用指针 p 正确地存取整型变量，因为分配时使用的是 sizeof(char)。

```
int *p;
p=(int *)malloc(40*sizeof(char));
```

- 有如下 C 语句，判断说法的正误：内存动态分配语句执行后能使用指针 p 正确地存取整型变量。

```
int *p;
p=(int *)malloc(40*sizeof(char));
```

- 有如下 C 语句，判断说法的正误：内存动态分配语句执行后能使用指针 p 正确地存取字符型变量。

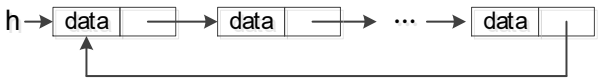
```
int *p;
p=(int *)malloc(40*sizeof(char));
```

- 有如下 C 语句，判断说法的正误：若不显式地释放动态分配的内存空间，当 p 因超出了作用域而被删除时，p 所指向的空间也将被系统自动释放。

```
int *p;
p=(int *)malloc(40*sizeof(char));
```

3. 程序完形填空

- 有如图所示的存储结构，每个结点两个域，data 是指向字符串的指针域，next 是指向结点的指针域，请填空完成此结构的类型定义。

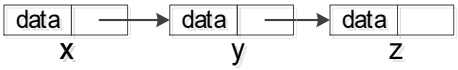



```
struct link
{
   （1）;
   （2）;
}*h;
```

- 设有以下定义：

```
struct node
{
   int data;
   struct node *next;
}x, y, z;
```

并有如图所示的链表结构，请写出删除 y 结点的赋值语句：_____。



- 以下程序建立了一个带有头结点的单链表，链表结点中的数据通过键盘输入，当输入数据为-1 时，表示输入结束（链表头结点*ph 的 data 域不放数据，表空的条件是 ph->next==NULL）。请填空：

```
#include <stdio.h>
#include <malloc.h>
struct list
{
   int data;
   struct list *next;
};
  （1）  creatlist()
{
   struct list *p, *q, *ph;
   int a;
```

```
    ph=(struct list *)malloc(sizeof(struct list));
    q=ph;
    printf("输入一个整数，输入-1 表示结束：");
    scanf("%d", &a);
    while(a!=-1)
    {
        p=(struct list *)malloc(sizeof(struct list));
        p->data=a;
        q->next=p;
        （2）=p;
        scanf("%d", &a);
    }
    q->next=NULL;
    return ph;
}
int main()
{
    struct list *head;
    head=creatlist();
    return 0;
}
```

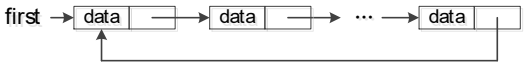
- 程序功能[1]：
  - ①从键盘输入姓名与成绩：利用动态链表存储；姓名为#时结束输入；②输出链表中的信息。
  - 请阅读以下源代码，并在“填空处”写出适当的 C 语言元素：

```
#include <stdio.h>
#include <malloc.h>
struct link
{
    char name[10];
    int   mark;
    struct link *next;
};
void insert(char *, int);
struct link *head=NULL;
main()
{
    char name[10];
    int mark;
    struct link *t;
    while (1)
    {
        scanf("%s%d", name, &mark);
```

[1] 19-20(1)2019 级“C 程序设计”期末考试 A 卷程序填空题-王红

```
        if(strcmp(name, "#")==0) break;
        （1）;
    }
    for(t=head; （2）)
        printf("<%s>: %d\n",  t->name,  t->mark);
}
void insert(char *name, int mark)
{
    struct link *p;
    p = （3）;
    strcpy(p->name, name);
    p->mark = mark;
    （4）;
    if(head != NULL) （5）;
    head = p;
}
```

- 下面 MIN3 函数的功能是：将如图所示的循环单链表 first（不带头结点，至少有 3 个以上的结点）中，自 first 指向的结点起，每 3 个相邻结点分为一组——假设结点个数为 3 的倍数。求每组结点数据域中值的和，返回其中最小的值。请填空。




```
struct node
{
   int data;
   struct node *link;
};
int MIN3(struct node *first)
{
   int m, m3;
   struct node *p=first;
   m=m3=p->data+p->link->data+ p->link->link->data;
   for(p=p->link->link->link; p!=first; p=  （1）  )
   {
      m=p->data+p->link->data+ p->link->link->data;
      if(  （2）  ) m3=m;
   }
   return m3;
};
```

- 下面程序的功能是建立一个“逆序”链表，即新生成的结点是链表的头，第 1 个生成的结点是链表“尾”（即第 1 个生成结点的 next 值是 NULL）。请填空。

```
#include <stdio.h>
#include <malloc.h>
struct link
{
```

```
    char name[10];
    int  mark;
    struct link *next;
};
void insert(char *, int);
struct link *head=NULL;
main()
{
    char name[10];
    int mark;
    struct  link * t;
    while (1)
    {
        scanf("%s %d",  name,  &mark);
        if ( strcmp(name, "#") == 0 )  break;
          （1）;
    }
    for (t=head;（2）)
        printf("<%s>: %d\n",  t->name,  t->mark);
    return 0;
}
void insert(char * name,  int mark)
{
    struct link * p;
    p = （3）;
    strcpy(p->name,  name);
    p->mark = mark;
  （4）;
    if ( head != NULL) （5）;
    head = p;
}
```

五、共用体、枚举、typedef

1. 单选

- 下面定义的变量 a 所占内存字节数是：_____。

```
union U
{
   char st[4];
   short int i;
   long l;
};
```

```
struct A
{
   short int c;
   union U u;
}a;
```

A) 4

B) 5

C) 6

D) 8

- 以下程序的输出结果是：_____。

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

A) 4

B) 5

C) 6

D) 0

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
   union
   {
      unsigned int n;
      unsigned char c;
   }u1;
   u1.c='A';
   printf("%c\n", u1.n);
   return 0;
```

```
}
```

A) 产生语法错误

B) 随机值

C) A

D) 65

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
typedef union
{
   long x[2];
   int y[4];
   char z[8];
}MYTYPE;
MYTYPE them;
int main()
{
   printf("%d\n", sizeof(them));
   return 0;
}
```

A) 32

B) 16

C) 8

D) 24

- 字符'0'的 ASCII 码的十进制数是 48，且数组的第 0 个元素在低位，则以下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
   union
   {
      short int i[2];
      long k;
      char c[4];
   }r, *s=&r;
   s->i[0]=0x39;
   s->i[1]=0x38;
   printf("%c\n", s->c[0]);
   return 0;
}
```

A) 39

B) 9

C) 38

D) 8

- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
    union
    {
        char s[2];
        short int i;
    }a;
    a.i=0x1234;
    printf("%x, %x\n", a.s[0], a.s[1]);
    return 0;
}
```

A) 12, 34
B) 34, 12
C) 12, 00
D) 34, 00

- 若有以下定义，则 sizeof(struct aa)的值是：_____。

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

A) 30
B) 32
C) 24
D) 22

- 设有以下定义，则下面不正确的叙述是：_____。

```
union data
{
    int i;
    char c;
    float f;
}a;
```

A) a 所占的内存长度等于成员 f 的长度
B) a 的地址和它的各成员地址都是同一地址
C) a 可以作为函数参数
D) 不能对 a 赋值，但可以在定义 a 时对它初始化

- 以下关于枚举类型的叙述，不正确的是：_____。

A) 枚举变量只能取对应枚举类型的枚举元素表中的元素
B) 可以在定义枚举类型时对枚举元素进行初始化
C) 枚举元素表中的元素有先后次序，可以进行比较
D) 枚举元素的值可以是整数或字符串

- 对下列枚举类型名的定义中正确的是：_____。

A) enum a={one, two, three};
B) enum a {one=9, two=-1, three};
C) enum a={"one", "two", "three"};
D) enum a {"one", "two", "three"};

- 设有如下枚举类型定义：

enum language {Basic=3, Assembly, Ada=100, COBOL, Fortran};
由枚举量 Fortran 的值为：_____。
A) 4
B) 7
C) 102
D) 103

- 正确的 k 值是：_____。

enum {a, b=5, c, d=4, e}k=e;
A) 3
B) 4
C) 5
D) 6

- 以下关于 typedef 的叙述，不正确的是：_____。

A) 用 typedef 可以声明各种类型名，但不能用来声明变
B) 用 typedef 可以增加新的数据类型
C) 用 typedef 是将已存在的类型就一个新的名称来代表
D) 使用 typedef 便于程序的通用

- 若有以下声明和定义：

typedef int *INTEGER;
INTEGER p, *q;
以下叙述正确的是：_____。
A) p 是 int 型变量
B) p 是基类型为 int 的指针变量
C) q 是基类型为 int 的指针变量
D) 程序中可用 INTEGER 代替 int 类型名

- 若要声明一个类型名为 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;

- 设有以下定义：

typedef union
{

```
    long i;
    short int k[5];
    char c;
}DATE;
struct date
{
    short int cat;
    DATE cow;
    double dog;
}too;
DATE max;
```

则下列语句的执行结果是：_____。

```
printf("%d,%d\n",sizeof(struct date),sizeof(max));
```

A) 24,12
B) 20,10
C) 36,24
D) 20,12

- 设有如下声明：

```
typedef struct
{
    int n;
    char c;
    double x;
}STD;
```

则以下选项中，能正确定义结构体数组并赋初值的语句是：_____。

A) STD tt[2]={{1, 'A', 62},{2, 'B', 75}};
B) STD tt[2]={1, "A", 62},{2, "B", 75};
C) STD tt[2]={{1, 'A', 2, 'B'}};
D) STD tt[2]={{1, "A", 62.5},{2, "B", 75.0}};

2. 判断

- 共用体类型是用关键字 union 声明的，其声明方式与声明结构体类型相同。
- 只能用第一个成员类型的值初始化一个共用体变量。
- 共用体所有成员共用的内存单元的大小为各成员需要占用的内存大小之和。
- 共用体所有成员都共用同一内存单元。

3. 填空

- 若有以下定义语句，则变量 w 在内存中所占的字节数是：_____。

```
union aa
{
    float x, y;
    char c[6];
```

```
};
struct st
{
    union aa v;
    float w[5];
    double ave;
}w;
```

4. 程序运行结果

- 下面程序的运行结果是：_____。

```
#include <stdio.h>
typedef union student
{
    char name[10];
    long sno;
    char sex;
    float score[4];
}STU;
int main()
{
    STU a[5];
    printf("%d\n", sizeof(a));
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct s
{
    char low;
    char high;
};
union m
{
    struct s byte;
    short int word;
};
int main()
{
    union m m1;
    m1.word=0x4567;
    printf("%x,%x,%x,", m1.word, m1.byte.high , m1.byte.low);
    m1.byte.low=0xff;
    printf("%x\n", m1.word);
```

```
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct s
{
    union
    {
        short int x;
        short int y;
    }c;
    short int a;
    short int b;
};
int main()
{
    struct s m;
    m.a=1; m.b=2;
    m.c.x=m.a*m.b;
    m.c.y=m.a+m.b;
    printf("%d,%d\n", m.c.x, m.c.y);
    return 0;
}
```

- 分析以下程序的输出结果。

```
#include <stdio.h>
struct byte
{
    short int x;
    char y;
};
union
{
    short int i[2];
    long j;
    char m[2];
    struct byte d;
}r, *s=&r;
int main()
{
    s->j=0x98765432;
    printf("%x,%x\n", s->d.x, s->d.y);
    return 0;
}
```

- 以下程序的运算结果是：_____。

```
#include <stdio.h>
int main()
{
   enum {a, b=5, c, d=4, e}k=c;
   printf("%d\n", k);
   return 0;
}
```

- 下面程序的运行结果是：_____。

```
#include <stdio.h>
int main()
{
      enum {a, b=5, c, d=4, e}k;
      int n;
      k=e;
      n=2*k;
      printf("%d\n", n);
      return 0;
}
```

- 以下程序的运算结果是：_____。

```
#include <stdio.h>
typedef int A[5];
int main()
{
   A a;
   int i;
   for(i=0; i<5; i++) a[i]=2*i;
   for(i=0; i<5; i++) printf("%d ", a[i]);
   printf("\n");
   return 0;
}
```

- 下面程序的运行结果是：_____。

```
#include <stdio.h>
#include <malloc.h>
typedef struct node
{
   int data;
   struct node *next;
}NODE;
int main()
{
   int a[]={2,4,1,3,5}, n=5, i;
   NODE *h=NULL, *p, *t;
   for(i=0; i<n; i++)
   {
```

```
        p=(NODE *)malloc(sizeof(NODE));
        p->data=a[i];
        if(h==NULL) h=t=p;
        else {t->next=p; t=p;}
    }
    p=h;
    for(; p!=t; p=p->next) printf("%d ", p->data);
    printf("%d\n", p->data);
    return 0;
}
```