

一、指针概念

1. 单选

- 变量的指针，其含义是指该变量的：____。
 - A) 值
 - B) 地址
 - C) 名
 - D) 一个标志
- 对于基类型相同的两个指针变量之间，不能进行的运算的是：____。
 - A) <
 - B) =
 - C) +
 - D) -
- 若有定义 `int *p1, *p2, m=2, n;`，以下均为正确赋值语句的选项是：____。
 - A) `p1=&m; p2=&p1;`
 - B) `p1=&m; p2=&n; *p1=*p2;`
 - C) `p1=&m; *p1=*p2;`
 - D) `p1=&m; *p2=*p1;`
- 设 `p1` 和 `p2` 均为指向一个 `int` 型数组的指针变量，`k` 为 `int` 型变量，则以下不能正确执行的赋值语句是：____。
 - A) `k=*p1+*p2;`
 - B) `p2=k;`
 - C) `p1=p2;`
 - D) `k=*p1*(p2);`
- 若已定义 `a` 为 `int` 型，则____是对指针变量 `p` 的正确定义和初始化。
 - A) `int *p=a;`
 - B) `int *p=*a;`
 - C) `int p=&a;`
 - D) `int *p=&a;`
- 假设整型变量 `a` 的值为 12，`a` 的地址为 2000，若欲使 `p` 为指向 `a` 的指针变量，则下面赋值正确的是：____。
 - A) `&a=3;`
 - B) `*p=12;`
 - C) `*p=2000;`
 - D) `p=&a;`
- 若有定义 `int n=2, *p=&n, *q=p;`，则以下非法赋值语句是：____。
 - A) `p=q;`
 - B) `*p=*q;`
 - C) `n=*q`
 - D) `p=n;`
- 若有定义 `int *p, m=5, n;`，则以下正确的 C 语句是：____。

A) `p=&n; scanf("%d", &p);`

B) `p=&n; scanf("%d", *p);`

C) `scanf("%d", &n); *p=n;`

D) `p=&n; *p=m;`

- 若有以下定义，则选项中错误的语句是：_____。

```
int a=4, b=3, *p, *q, *w;
```

```
p = &a; q=&b; w=q; q=NULL;
```

A) `*q=0;`

B) `w=p;`

C) `*p=a;`

D) `*p=*w;`

- 如下程序段，执行该程序段后，a 的值是：_____。

```
int *p, a=10, b=1;
```

```
p = &a;
```

```
a = *p+b;
```

A) 12

B) 11

C) 10

D) 编译出错

- 以下程序段编译时出现错误，则下面叙述中正确的是：_____。

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char a, b, c, *d;
```

```
    a='\';
```

```
    b='\xbc';
```

```
    c='\0xab';
```

```
    d= "017";
```

```
    printf("%c%c%c%c\n", a, b, c, *d);
```

```
    return 0;
```

```
}
```

A) 程序中只有 `a='\'`; 语句错误

B) `b='\xbc';` 语句错误

C) `d= "017";` 语句错误

D) `a='\'`; 和 `c='\0xab';` 语句都错误

- 执行以下程序后，a 和 b 的值分别是：_____。

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

```
int main()
```

```
{
```

```
    int a, b, k=4, m=6, *p1=&k, *p2=&m;
```

```
    a=p1==&m;
```

```
    b=(-*p1)/(*p2)+7;
```

```
    printf("%d,%d\n", a,b);
```

```
return 0;  
}
```

- A) -1,5
- B) 1,6
- C) 0,7
- D) 4,10

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
int main()  
{  
    int k=3, j=4;  
    int *p=&k, *q=&j;  
    (*p)--;  
    j=k;  
    (*q)--;  
    printf("%d,%d\n", k, j);  
    return 0;  
}
```

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

2. 判断

- 语句 `int *p;` 中的 `*p` 的含义为取值。
- 指针变量和普通变量相似，其值是可变的。
- 所有指针变量都用于存入地址值，所以指针变量与目标类型无关。
- `int i, *p=&i;` 是正确的 C 语言定义。
- 指针的变量可以初始化为 NULL 或一个地址。
- 指针的变量可以初始化为任一整数常量。
- `int a=5, *p; *p=&a;`，则 `p` 的值为 5。
- 两个指针相加没有意义。
- 两指针变量相减所得之差是两个指针所指元素之间相差的元素个数。
- 设指针变量 `p` 已指向一个有效地址，则表达式 `&*p==p` 的结果为真。
- 若 `p` 是一个指针变量，表达式 `*p++` 的运算顺序为先自增后取值。
- 若 `p` 是一个指针变量，执行语句 `*--p;` 后，`p` 指向的内存单元被改变。
- 若 `p` 是一个指针变量，执行语句 `(*p)++;` 之后，`p` 指向的内存单元的值被改变。
- 若 `p` 是一个指针变量，执行表达式 `(*p)++` 后，`p` 的指向改变了。

3. 简答

- 用变量 `a` 给出以下定义：
 - (1) 一个整型数

- (2) 一个指向整型数的指针。
- (3) 一个指向指针的指针，它指向的指针是指向一个整型数。
- (4) 一个有 10 个整型数的数组。
- (5) 一个有 10 个指针的数组。
- (6) 一个指向有 10 个整型数数组的指针。
- (7) 一个指向函数的指针，该函数有一个整型参数并返回一个整型数。
- (8) 一个有 10 个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数。
- 有定义：`float *p[4];`，则下面叙述中哪些是正确的？
 - (1) 此定义错误，形如 `char *p[4];` 的定义才是正确的。
 - (2) 此定义正确，`p` 是指向一维实型数组的指针变量，而不是指向单个实型变量的指针变量。
 - (3) 此定义错误，C 语言中不允许类似定义。
 - (4) 此定义正确，定义了一个指针数组。
- 解释下面 C 定义语句：
 - (1) `int **a;`
 - (2) `long (*b)[3];`
- 指出 `int *p[3];` 和 `int (*p)[3];` 两个定义之间的差别。
- 在定义语句 `float f[3][4], *p1, *p2[k];` 中，若 `k` 是 0 到 2 之间的常量（包括 0 和 2），则下列赋值语句中哪些是错误的？
 - (1) `p2=f;`
 - (2) `p1=f[k];`
 - (3) `p2[k]=f[k];`
 - (4) `p1=&f[0][0];`
 - (5) `p1=p2[k];`

4. 程序运行结果

- 以下程序输出的结果是：_____。

```
#include <stdio.h>
int main()
{
    int i, *p;
    p=&i;
    *p=8;
    printf("i=%d\n", i);
    return 0;
}
```

- 以下程序输出的结果是：_____。

```
#include <stdio.h>
int main()
{
    int *p1, *p2, *p;
    int a=10, b=12;
```

```

    p1=&a; p2=&b;
    if(a<b)
    {
        p=p1; p1=p2; p2=p;
    }
    printf("%d,%d,", *p1, *p2);
    printf("%d,%d\n", a, b);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int m[12], k;
    int *p[3], sum=0;
    for(k=0; k<12; k++)
    {
        m[k]=2*k;
        if(k<3) p[k]=m+2*k*k;
    }
    for(k=0; k<3; k++) sum+=*(p+k)[4-k];
    printf("sum=%d\n", sum);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int **k, *j, i=100;
    j=&i; k=&j;
    printf("%d\n", **k);
    return 0;
}

```

- 下面程序的输出结果是：_____。

```

#include <stdio.h>
int main()
{
    int a[]={2, 6, 10, 14, 18};
    int *ptr[]={&a[0], &a[1], &a[2], &a[3], &a[4]};
    int **p, i;
    for(i=0; i<5; i++) a[i]=a[i]/2+a[i];
    p=ptr;
    printf("%d ", *(*p+2));
    printf("%d\n", *(*++p));
}

```

```

    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int a[]={1,3,5,7};
    char *p[3]={a+2,a+1,a};
    char **q=p;
    printf("%d\n",*(p[0]+1)+**(q+2));
    return 0;
}

```

二、利用指针处理数组

1. 单选

- 以下程序调用 findmax 函数以返回数组中的最大值，则程序下划线处应填入的选项是：_____。

```

#include <stdio.h>
findmax(int *a, int n)
{
    int *p, *s;
    for(p=a, s=a; p-a<n; p++)
        if(_____) s=p;
    return *s;
}
int main()
{
    int x[5]={12,21,13,6,18};
    printf("%d\n", findmax(x, 5));
    return 0;
}

```

- A) p>s
- B) *p>*s
- C) a[p]>a[s]
- D) p-a>p-s

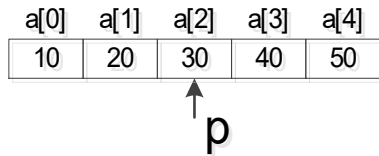
- 以下合法的数据定义是：_____。

- A) int s[] = "china";
- B) int s[2] = {0,2,4};
- C) char s = "china";
- D) char s[] = {"0,1,2,3,4,5"};

- 若有定义 `int a[]={1,3,5,7,9,11}, *ptr=a;`, 则能够正确地引用该数组元素的是: ____。

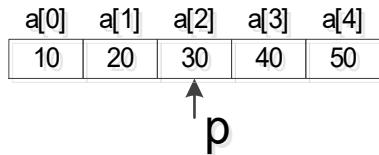
A) a
B) *(ptr--)
C) a[6]
D) *(--ptr)

- 已知指针 `p` 的指向如图所示, 则执行 `*--p` 的返回值是: ____。



A) 30
B) 20
C) 19
D) 29

- 已知指针 `p` 的指向如图所示, 则执行 `*--p` 后 `*p` 的返回值是: ____。



A) 30
B) 20
C) 19
D) 29

- 以下程序的输出结果是: ____。

```
#include <stdio.h>
int main()
{
    int c[]={1,7,12};
    int *p=c+1;
    printf("%d\n", *p++);
    return 0;
}
```

A) 2
B) 7
C) 8
D) 12

- 以下程序的输出结果是: ____。

```
#include <stdio.h>
int main()
{
    int a[]={2,4,6,8,10}, y=1, x, *p;
    p=&a[1];
```

```

    for(x=0; x<3; x++) y+=*(p+x);
    printf("%d\n", y);
    return 0;
}

```

- A) 17
- B) 18
- C) 19
- D) 20

- 有如下定义，则数值 9 对应的表达式是：_____。

```
int a[10]={1,2,3,4,5,6,7,8,9,10}, *p=a;
```

- A) *p+9
- B) *(p+8)
- C) *p+=9
- D) p+8

- 以下程序的输出结果是：_____。

```

#include <stdio.h>
int main()
{
    int x[8]={8,7,6,5,10,0},*s;
    s=x+3;
    printf("%d\n", s[2]);
    return 0;
}

```

- A) 10
- B) 0
- C) 5
- D) 6

- 以下程序的输出结果是：_____。

```

#include <stdio.h>
int main()
{
    int a[5] = {1,2,3,4,5};
    int *ptr=(int *)(&a+1);
    printf("%d, %d\n", *(a+1), *(ptr-1));
    return 0;
}

```

- A) 1,2
- B) 2,5
- C) 1,5
- D) 2,1

- 有如下定义，并且 $0 \leq i \leq 9$ ，则对 a 数组元素的引用错误的是：_____。

```
int a[]={0,1,2,3,4,5,6,7,8,9}, *p=a, i;
```

- A) a[p-a]
- B) *(&a[i])

- C) p[i]
D) *(*a+i))
- 若有语句 `int x[3][4];`，则以下关于 `x`、`*x`、`x[0]`、`&x[0][0]` 的正确描述是：_____。
 - A) `x`、`*x`、`x[0]`、`&x[0][0]` 均表示元素 `x[0][0]` 的地址
 - B) 只有 `*x`、`x[0]`、`&x[0][0]` 表示的是元素 `x[0][0]` 的地址
 - C) 只有 `x[0]`、`&x[0][0]` 表示的是元素 `x[0][0]` 的地址
 - D) 只有 `&x[0][0]` 表示的是元素 `x[0][0]` 的地址
 - 若有定义 `int (*p)[3];`，则以下_____是正确的叙述。
 - A) `p` 是一个指针数组
 - B) `p` 是一个指针，它只能指向一个包含 3 个 `int` 类型元素的二维数组
 - C) `p` 是一个指针，它可以指向一个一维数组中的任一个元素
 - D) `(*p)[3]` 与 `*p[3]` 等价
 - 以下_____是一个指向二维整数数组的指针的定义。
 - A) `int (*ptr)[3]`
 - B) `int *ptr[3]`
 - C) `int (*ptr[3])`
 - D) `int ptr[3]`
 - 若有定义语句 `int i, x[3][4];`，则不能将 `x[1][1]` 的值赋给变量 `i` 的语句是：_____。
 - A) `i=*(*(x+1)+1);`
 - B) `i=x[1][1];`
 - C) `i=*(x+1);`
 - D) `i=*(x[1]+1);`
 - 有下面定义语句，并且 $0 \leq m \leq 1, 0 \leq n \leq 2$ ，则_____是对数组元素的正确引用。


```
int i, a[2][3]={1,3,5,7,9,11};
int m, n;
```

 - A) `a[m]+n`
 - B) `*(a+5)`
 - C) `*(*(a+m)+3)`
 - D) `*(*(a+m)+n)`
 - 以下程序的输出结果是：_____。


```
#include <stdio.h>
int main()
{
    int a[3][3], *p, i;
    p=&a[0][0];
    for(i=0; i<9; i++) p[i]=i+1;
    printf("%d\n", a[1][2]);
    return 0;
}
```

 - A) 3
 - B) 6
 - C) 9
 - D) 随机数
 - 以下程序的输出结果是：_____。

```
#include <stdio.h>

int 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]);
    return 0;
}
```

- A) 7
- B) 6
- C) 8
- D) 无确定值

- 已知 `int a[4][3]={1,2,3,4,5,6,7,8,9,10,11,12}; int (*ptr)[3]=a, *p=a[0];`, 则以下能够正确表示数组元素 `a[1][2]` 的表达式是_____

- A) `*((ptr+1)[2])`
- B) `*(*(p+5))`
- C) `(*ptr+1)+2`
- D) `*(*(a+1)+2)`

- 已知 `int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}; int *p=a[0]; p+=6;`, 则以下与 `*p` 含义相同的是_____

- A) `*(a+6)`
- B) `*(&a[0]+6)`
- C) `*(a[1]+2)`
- D) `*a[1]+2`

- C 语言的定义语句 `char *a[5]` 的含义是：_____。
 - A) `a` 是一个数组，其数组的每一个元素是指向字符的指针
 - B) `a` 是一个指针，指向一个数组，数组的元素为字符型
 - C) A 和 B 均不对，但它是 C 语言正确的语句
 - D) C 语言不允许这样定义语句

- 若有定义 `int a[4][10], *p, *q[4];`, 且 $1 < i < 4$, 则_____是错误的赋值。

- A) `p=a`
- B) `q[i]=a[i]`
- C) `p=a[i]`
- D) `q[i]=&a[2][0]`

- 以下程序的输出结果是：_____。

```
#include <stdio.h>

void fun(char **m)
{
    ++m;
    printf("%s\n", *m);
}
```

```
int main()
{
    char *a[]={ "BASIC", "FOXPRO", "C"};
    fun(a);
    return 0;
}
```

- A) BASIC
- B) ASIC
- C) FOXPRO
- D) C

2. 判断

- 数组名和指针变量是相互等价的。
- 数组名实际上是此数组的首元素的首地址，所以数组名相当于一个指针变量。
- 某函数的形参为一个数组，则调用此函数时只能将数组名作为对应的实参。
- 假设有 `int a[10], *p;`，则 `p=&a[0]` 与 `p=a` 等价。
- 有定义 `int a[10]={1,2,3,4,5,6,7,8,9,10}, *p=a;`，则数值为 9 的表达式是 `*(p+8)`。
- 对于二维数组 `a`，`*(a[i]+j)` 与 `a[i][j]` 的含义相同。
- `int (*p)[4]` 表示 `p` 是一个指针数组，它包含 4 个指针变量元素。
- 若定义数组 `a[2][3]`，则 `a+1` 和 `*(a+1)` 完全等价。
- 若定义数组 `a[2][3]`，则 `++a` 和 `a[1]` 完全等价。
- 若有定义 `char s[3][4];`，则下面对数组元素 `s[i][j]` 的引用是否正确： `*(s+i)[j]`
- 若有定义 `char s[3][4];`，则下面对数组元素 `s[i][j]` 的引用是否正确： `*(&s[0][0]+4*i+j)`
- 若有定义 `char s[3][4];`，则下面对数组元素 `s[i][j]` 的引用是否正确： `*((s+i)+j)`
- 若有定义 `char s[3][4];`，则下面对数组元素 `s[i][j]` 的引用是否正确： `*(*(s+i)[j])`
- `*(*(s+i)[j])` 等价于 `***s+i+j`，错误的表达方式。
- 若有以下程序段：


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

 则对 `a` 数组元素 `a[i][j]` ($0 \leq i \leq 3, 0 \leq j \leq 4$) 的正确引用的选项是： `p+1`
- 若有以下程序段：


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

 则对 `a` 数组元素 `a[i][j]` ($0 \leq i \leq 3, 0 \leq j \leq 4$) 的正确引用的选项是： `*(p+3)`
- 若有以下程序段：


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

 则对 `a` 数组元素 `a[i][j]` ($0 \leq i \leq 3, 0 \leq j \leq 4$) 的正确引用的选项是： `*(p+1)+3`
- 若有以下程序段：


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

 则对 `a` 数组元素 `a[i][j]` ($0 \leq i \leq 3, 0 \leq j \leq 4$) 的正确引用的选项是： `*(p+2)`

- 设变量定义为 `int *p[3], a[3];`, 则表达式 `p=&a[0]`、`*p=**a`、`p[0]=a`、`**p=a` 语句都是正确的。

3. 填空

- 有一个数组 `a` 含有 5 个元素, 若 `p` 已指向存储单元 `a[1]`。通过指针 `p` 给指针变量 `s` 赋值, 使 `s` 指向最后一个存储单元 `a[4]` 的语句是: _____。
- 有一个数组 `a`, 若指针 `s` 指向存储单元 `a[2]`, 指针 `p` 指向存储单元 `a[0]`, 表达式 `s-p` 的值是: _____。
- 若有定义 `int a[10]={23,54,10,33,47,98,72,80,6}, *p=a;`, 则不移动指针 `p`, 且通过指针 `p` 引用值为 98 的数组元素的表达式是: _____。

4. 简答

- 设有以下程序:

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

问执行此程序, 如果第一个 `printf` 输出的值是 200, 则第二个 `printf` 输出是多少?

5. 程序运行结果

- 以下程序的输出结果是: _____

```
#include <stdio.h>
int main()
{
    int a[]={1, 3, 5, 7, 9};
    int x, y, *ptr;
    y=1;
    ptr=&a[1];
    for(x=0; x<3; x++) y*=*(ptr+x);
    printf("%d\n", y);
    return 0;
}
```

- 以下程序的输出结果是: _____

```
#include <stdio.h>
int main()
```

```

{
    int x[]={1,2,3}, s, i, *p;
    s=1; p=x;
    for(i=0; i<3; i++) s*=*(p+i);
    printf("%d\n", s);
    return 0;
}

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()
{
    int a[2][3]={ {1,2,3}, {4,5,6} };
    int m, *ptr;
    ptr=&a[0][0];
    m=(*ptr)*(*(ptr+2))*(*(ptr+4));
    printf("%d\n", m);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int a[3][4]={1,2,3,4,5,6,7,8}, *pa=&a[0][0];
    printf("%d\n", *(pa+6));
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int a[3][4]={1,2,3,4,5,6,7,8};
    printf("%d\n", *(a+1)[1]);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int a[3][4]={1,2,3,4,5,6,7,8,9,10}, (*pa)[4]=a;
    printf("%d\n", (*(pa+1))[2]);
    return 0;
}

```

6. 程序完形填空

- 下面程序的功能是：将无符号八进制数字构成的字符串转换为十进制整数，例如，输入的字符串为 556，则输出十进制整数 366。请填空：

```
#include <stdio.h>
int main()
{
    char *p, s[6];
    int n;
    p=s;
    gets(p);
    n=*p-'0';
    while(_____ != '\0') n=n*8+*p-'0';
    printf("%d\n", n);
    return 0;
}
```

- 以下程序实现从 10 个数中找出最大值和最小值，请填空。

```
#include <stdio.h>
int main()
{
    int a[]={6,1,5,2,3,9,10,4,8,7}, *q;
    int max, min;
    max=min=*a;
    for(q= (1) ; (2) ; q++)
        if( (3) ) max=*q;
        else if( (4) ) min=*q;
    printf("max=%d, min=%d\n", max, min);
    return 0;
}
```

- 以下程序通过指向的数组 a[3][4] 元素的指针将其内容按 3 行 4 列的格式输出，请在调用 printf 的语句中填入适当的参数，使之通过指针 p 将数组元素按要求输出。

```
#include <stdio.h>
int main()
{
    int a[3][4]={ {1,2,3,4}, {5,6,7,8}, {9,10,11,12} }, *p=&a[0][0];
    int i, j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++) printf("%3d", _____);
        printf("\n");
    }
    return 0;
}
```

- 以下程序通过指向的数组 a[3][4] 元素的指针将其内容按 3 行 4 列的格式输出，请在调用 printf 的语句中填入适当的参数，使之通过指针 p 将数组元素按要求输出。

```
#include <stdio.h>
int main()
{
    int a[3][4]={ {1,2,3,4},{5,6,7,8},{9,10,11,12}}, (*p)[4]=a;
    int i, j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++) printf("%3d", _____);
        printf("\n");
    }
    return 0;
}
```

三、利用指针处理字符串

1. 单选

- 以下合法的定义是：_____。
 - A) str[]={“china”};
 - B) char *p=“china”;
 - C) char *p; strcpy(p, “china”);
 - D) char str[13]; str[]=“china”;
- 设有定义 char *st=“how are you”;，下列程序段中正确的是：_____。
 - A) char a[11], *p; strcpy(p=a+1, &st[4]);
 - B) char a[11]; strcpy(++a, st);
 - C) char a[11]; strcpy(a, st);
 - D) char a[], *p; strcpy(p=&a[1], st+2);
- 已有定义 char *st=“how are you”;，下列语句会引发“编译”错误的是：_____。
 - A) char a[11], *p; strcpy(p=a+1, &st[4]);
 - B) char a[11]; strcpy(++a, st);
 - C) char a[11]; strcpy(a, st);
 - D) char a[11], *p; strcpy(p=&a[1], st+2);
- 以下程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
    char a[]="Language", b[]="programe", *p1, *p2;
    int k;
    p1=a; p2=b;
    for(k=0; k<7; k++)
        if(*(p1+k) == *(p2+k)) printf("%c", *(p1+k));
}
```

```
return 0;  
}
```

A) gae

B) ga

C) Language

D) 有语法错误

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
#include <string.h>  
int main()  
{  
    char b1[8]="abcdefg", b2[8], *pb=b1+3;  
    while(--pb>=b1) strcpy(b2, pb);  
    printf("%d\n", strlen(b2));  
    return 0;  
}
```

A) 8

B) 3

C) 1

D) 7

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
#include <string.h>  
int main()  
{  
    char *p="abcde\0fghijk\0";  
    printf("%d\n", strlen(p));  
    return 0;  
}
```

A) 12

B) 15

C) 6

D) 5

- 以下程序的输出结果是：_____。

```
#include <stdio.h>  
int main()  
{  
    char str[]="xyz", *ps=str;  
    while(*ps) ps++;  
    for(ps--; ps-str>=0; ps--) puts(ps);  
    return 0;  
}
```

A) yz

xyz

- xyz
- B) z
- yz
- xyz
- C) z
- yz
- D) x
- xy

- 下面程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
    char str[2][5]={"6937", "8254"}, *p[2];
    int i, j, s=0;
    for(i=0; i<2; i++) p[i]=str[i];
    for(i=0; i<2; i++)
        for(j=0; p[i][j]>'\0'; j+=2) s=10*s+p[i][j]-'0';
    printf("%d\n", s);
    return 0;
}
```

- A) 69825
- B) 63825
- C) 6385
- D) 693825

- 下面程序的输出结果是：_____。

```
#include <stdio.h>
int main()
{
    char s[][10]={"one", "two", "three"}, *p;
    p=s[1];
    printf("%c, %s\n", *(p+1), s[0]);
    return 0;
}
```

- A) n, two
- B) t, one
- C) w, one
- D) o, two

- 下面程序的输出结果是：_____。

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[][10]={"Hello", "Google"};
    char *p=str[0];
```

```
printf("%d\n", strlen(++p));
return 0;
}
```

- A) 4
- B) 5
- C) 6
- D) 7

2. 判断

- `char *p="girl"`的含义是定义字符型指针变量 `p`，`p` 的值是字符串“girl”。
- `char *s="C Language"`表示 `s` 是一个指向字符串的指针变量，把字符串的首地址赋予 `s`。
- 判断语句的正误：`char *p="Good morning"`;
- 判断语句的正误：`char *p, p="Good morning"`;
- 判断语句的正误：`char str[]="Good morning"`;
- 判断语句的正误：`char str[20]; str="Good morning"`;

3. 简答

- 下面程序的功能是在 5 个字符串中求最小字符串并输出结果，请指出程序中的错误：

```
#include <stdio.h>
#include <string.h>
int main()
{
    char *name[5]={"Windows", "Word", "Excel", "Foxpro", "Wisual Basic"};
    char temp;
    int i;
    temp=name[0];
    for(i=1; i<5; i++)
        if(temp>(*name[i])>0) temp=name[i];
    printf("%s\n", *temp);
    return 0;
}
```

4. 程序运行结果

- 以下程序的输出结果是：_____

```
#include <stdio.h>
int main()
{
    char *ptr1, *ptr2;
    ptr1=ptr2="abcde";
```

```

        while(*ptr2 != '\0') putchar(*ptr2++);
        while(--ptr2 >= ptr1) putchar(*ptr2);
        putchar('\n');
        return 0;
    }

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()
{
    char a[]="english", *ptr=a;
    while(*ptr)
    {
        printf("%c", *ptr-'a'+'A');
        ptr++;
    }
    printf("\n");
    return 0;
}

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()
{
    char *p="abcdefghijklmnopq";
    while(*p++ != 'e');
    printf("%c\n", *p);
    return 0;
}

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()
{
    char *a="PROGRAM";
    char b[]="program";
    int i=0;
    printf("%c%s, ", *a, b+1);
    while(putchar(*(a+i))) i++;
    printf(", i=%d, ", i);
    while(--i) putchar(*(b+i));
    printf(", %s\n", &b[3]);
    return 0;
}

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()

```

```

{
    char s[]="9876", *p;
    for(p=s; p<s+2; p++) printf("%s ", p);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    char s[]={"abcdef"};
    char *p=s;
    *(p+2)+=3;
    printf("%c,%c\n", *p, *(p+2));
}

```

- 以下程序的输出结果是： _____

```

#include <stdio.h>
int main()
{
    char *p="abcdefgh", *r;
    int *q;
    q=(int *)p;
    q++;
    r=(char *)q;
    printf("%s\n", r);
    return 0;
}

```

- 下面程序的输出结果是： _____。

```

#include <stdio.h>
int main()
{
    char *alpha[6]={"ABCD", "EFGH", "IJKL", "MNOP", "QRST", "UVWX"};
    char **p;
    int i;
    p=alpha;
    for(i=0; i<4; i++) printf("%s", p[i]);
    printf("\n");
    return 0;
}

```

- 下面程序的输出结果是： _____。

```

#include <stdio.h>
int main()
{
    int i;
    char **p, *a[]={"dog", "cat", "chook"};

```

```

for(p=a,i=0;i<3;i++)
    printf("%s,%c\n",*(p+i),*(*(p+i)+i));
return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int main()
{
    int i;
    char **p;
    char *greeting[]={"Hello","Good morning","How are you"};
    p=greeting;
    for(i=0;i<=2;i++) printf("greeting[%d]=%s\n",i,greeting[i]);
    while(**p!='\0') printf("%s\n",*p++);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
#define P(x) printf("%s", x)
int main()
{
    char *c[]={"You can make statement","for the topic","The sentences","Hou about"};
    char **p[]={c+3, c+2, c+1, c};
    char ***pp=p;
    P(**++pp);
    P(*--*++pp+3);
    P(*pp[-2]+3);
    P(pp[-1][-1]+3);
    printf("\n");
    return 0;
}

```

5. 程序完形填空

- 下面 `mystrlen` 函数的功能是计算指针 `str` 指向的字符串长度，并作为函数值返回。请填空：

```

int mystrlen(char *str)
{
    char *p;
    for(p=str; (1) !='\0'; p++);
    return (2);
}

```

- 函数 `match_ch(str,ch)` 完成在字符串 `str` 中寻找与 `ch` 匹配的字符，若存在匹配，则返回 `ch` 在 `str` 中第 1 次出现；否则，返回 -1。

```

int match_ch(char *str, char ch)
{
    char *p;
    for( (1) ; *p!='\0'; p++)
    {
        if( (2) ) break;
    }
    if(*p!='\0')
        return p-str;
    else
        return -1;
}

```

- 函数 match(s,t) 完成在字符串 s 中寻找与字符串 t 匹配的子串，若存在这样一个匹配，则返回子串 t 在字符串 s 中第 1 次出现的下标；否则，返回 -1。其中，字符指针 *b 始终指向 s 的第 1 个字符。

```

int match(char *s, char *t)
{
    char *b=s, *p, *r;
    for( (1) )
    {
        for(p=s, r=t; *r!='\0' && p!='\0' && *p==*r; p++, r++);
        if( (2) ) return(s-b);
    }
    return -1;
}

```

- count 函数的功能是统计子串 substr 在主串 str 中出现的次数。请填空。

```

int count(char *str, char *substr)
{
    int i, j, k, num=0;
    for(i=0; (1); i++)
    {
        for( (2), k=0; substr[k]==str[j]; k++, j++)
            if(substr[ (3) ]=='\0')
            {
                num++;
                break;
            }
    }
    return num;
}

```

- 函数 replace() 将字符串 old 中出现的所有 sub 子串都以 rpl 子串替代，形成新的字符串 news。例如 old 为 “uvwabuv uvuvwx uvxw”，sub 为“uvw”，rpl 为“abcd”，news 的最终结果为“abcdabuv uvabcdx uvxw”。请在下面程序段空白处填上适宜的内容：

```

void replace(char *old, char *sub, char *rpl, char *news)

```

```
{
    char *s1, *s2;
    while( (1) )
    {
        for(s1=old, s2=sub; *s1!='\0' && *s2!='\0' && (2); s1++, s2++);
        if(*s2 != '\0') //在 old 中未找到 sub
            *news++= (3) ;
        else
        {
            //在 old 中找到了 sub
            for(s2=rpl; *s2!='\0'; s2++) *news++= (4);
            (5);
        }
    }
    *news='\0';
}
```