

一、选择题

1. 设有定义 `int x=0, *p=&x;`, 则语句 `printf("%d\n", *p);` 的输出结果是_____。

- A. 随机值 B. 0 C. x 的地址 D. p 的地址

2. 设有定义 `int a, *pa=&a;`, 以下 `scanf` 语句中能正确为变量 `a` 读入数据的是_____。

- A. `scanf("%d", pa);` B. `scanf("%d", a);`
C. `scanf("%d", &pa);` D. `scanf("%d", *pa);`

3. 设有如下定义和赋值语句, 其指向关系如图 7.1 所示。若要将指针的指向关系改变为如图 7.2 所示, 可选用的赋值语句是_____。

```
int a=10, b=20, *p1, *p2;
p1=&a; p2=&b;
```

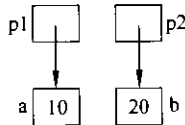


图 7.1 原来的指向关系

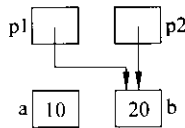


图 7.2 改变后的指向关系

- A. `*p1 = *p2;` B. `p1 = p2;` C. `p1 = *p2;` D. `*p1 = p2;`

4. 设有定义 `int a[5];`, 则 `a` 数组中首元素的地址可以表示为_____。

- A. `&a` B. `a+1` C. `a` D. `&a[1]`

5. 设有定义 `int a[10], *p=a;`, 则 `*(p+5)` 表示_____。

- A. 元素 `a[5]` 的地址 B. 元素 `a[5]` 的值
C. 元素 `a[6]` 的地址 D. 元素 `a[6]` 的值

6. 设有定义 `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_____。

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

7. 设有定义 `int i, a[10], *p;`, 则合法的赋值语句是_____。

- A. `p=100;` B. `p=a[5];` C. `p=a[2]+2;` D. `p=a+2;`


```

{
    int i,s=0,t[]={1,2,3,4,5,6,7,8,9};
    for(i=0;i<9;i+=2)
        s+=*(t+i);
    printf("s=%d\n",s);
    return 0;
}

```

4. 写出下列程序的执行结果。

```

#include<stdio.h>
int main()
{
    int array[8]={12,25,33,36,51,53,60,99};
    int *p;
    for(p=array;p<array+8;p++)
    {
        if(*p%10<5)
            *p=*p-*p%10;
        else
            *p=*p+10-*p%10;
    }
    for(p=array;p<array+8;p++)
        printf("%4d",*p);
    return 0;
}

```

5. 写出下列程序的执行结果。

```

#include<stdio.h>
int main()
{
    char ch[]="123ab56",*p=ch;
    int s=0;
    for(p=ch;*p>='0'&&*p<='9';p++)
        s=10*s+*p-'0';
    puts(ch);
    printf("s=%d\n",s);
    return 0;
}

```

6. 写出下列程序的执行结果。

```

#include<stdio.h>
void fun(int *x,int *y)
{
    printf("%d %d\n",*x,*y);
}

```

```

    * x=3;
    * y=4;
}
int main()
{
    int x=1,y=2;
    fun(&y,&x);
    printf("%d %d\n",x,y);
    return 0;
}

```

7. 写出下列程序执行时输入 3 5 的输出结果。

```

#include "stdio.h"
void fun(int * x,int * y)
{
    * x= * x+ * y;
    * y= * x- * y;
    * x= * x- * y;
}
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("a=%d,b=%d\n",a,b);
    fun(&a,&b);
    printf("a=%d,b=%d\n",a,b);
    return 0;
}

```

8. 写出下列程序的执行结果。

```

#include "stdio.h"
void swap(char * x,char * y)
{
    char * t;
    t=x;x=y;y=t;
}
int main()
{
    char ch1='a',ch2='b';
    printf("%c,%c\n",ch1,ch2);
    swap(&ch1,&ch2);
    printf("%c,%c\n",ch1,ch2);
    return 0;
}

```

9. 写出下列程序的执行结果。

```
#include<stdio.h>
int sum (int * array,int n);
int main(void)
{
    int b[5]={1,2,3,4,5};
    printf("%d\n",sum(b+1,3));
    return 0;
}
int sum (int * array,int n)
{
    int i,s=0;
    for(i=0;i<n;i++)
        s+= * (array+i);
    return(s);
}
```

10. 写出下列程序的执行结果。

```
#include "stdio.h"
void sub(int x,int y,int * z)
{
    * z=y-x;
}
int main()
{
    int a,b,c;
    sub(10,5,&a);
    sub(7,a,&b);
    sub(a,b,&c);
    printf("%d,%d,%d\n",a,b,c);
    return 0;
}
```

11. 写出下列程序的执行结果。

```
#include "stdio.h"
void change(int k[])
{
    k[0]=k[5];
}
int main()
{
    int x[10]={1,2,3,4,5,6,7,8,9,10},n=0;
    while(n<=4)
```

```

    {
        change(&x[n]);
        n++;
    }
    for(n=0;n<5;n++)
        printf("%3d",x[n]);
    printf("\n");
    return 0;
}

```

12. 写出下列程序的执行结果。

```

#include<stdio.h>
#include<string.h>
int main()
{
    char s[20]="abcd";
    char * sp=s;
    sp++;
    puts(strcat(sp,"ABCD"));
    return 0;
}

```

13. 分两次运行下列程序,分别输入“book”“hello world”,两次运行的结果分别是什么?(注: hello world 中间有一个空格符。)

```

#include<stdio.h>
int fun(char * s);
int main()
{
    char string[80];
    gets(string);
    printf("%d\n",fun(string));
    return 0;
}
int fun(char * s)
{
    int n;
    for(n=0; * s!='\0';s++)
        n++;
    return n;
}

```

14. 写出下列程序的执行结果。

```

#include "stdio.h"
int main()

```

```

{
    int *larger(int *a,int *b);
    int x=10,y=20;
    int *p;
    p=larger(&x,&y);
    printf("结果是%d\n", *p);
    return 0;
}
int *larger(int *a,int *b)
{
    if(*a>*b)
        return a;
    else
        return b;
}

```

15. 写出下列程序的执行结果。

```

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

```

三、程序填空题

1. 以下程序的功能是判断输入的字符串是否是回文(顺读和倒读都一样的字符串称为回文,如 level),请填空。

```

#include "stdio.h"
#include<string.h>
int main()
{
    char s[81], *p1, *p2;
    int n;
    gets(s);
    n=strlen(s);

```

```

p1=_____;
p2=_____;
while(_____)
{
    if(*p1!=*p2) break;
    else {p1++;_____;}
}
if(p1<p2) printf("No\n");
else printf("Yes\n");
return 0;
}

```

2. 以下函数 fun 的功能是返回 str 所指字符串中以形参 c 中字符开头的后续字符串的首地址。例如,假设 str 所指字符串为“Hello!”,形参变量 c 中的字符为'e',则函数返回字符串“ello!”的首地址。若 str 所指字符串为空串或不包含 c 中的字符,则函数返回 NULL。请填空。

```

char * fun(char * str,char c)
{
    int n=0;
    char * p=str;
    if(p!=NULL)
        while(p[n]!=c&&p[n]!='\0')
            n++;
    if(p[n]=='\0') return NULL;
    return(______);
}

```

四、编程题

1. 已知 char a;int x;float p,q;,并使用如下语句对变量进行了赋值: a='A';x=125;p=10.25;q=18.75;。编写程序显示变量 a、x、p、q 的值及其地址。

[输出示例]

```

A is stored at addr 1245052.
125 is stored at addr 1245048.
10.25 is stored at addr 1245044.
18.75 is stored at addr 1245040.

```

[提示] 使用运算符 & 获得变量的地址,地址输出采用 %u 格式(因为内存地址是无符号的整数)。

[思考] 你的执行结果和输出示例完全一致吗?为什么?

2. 编写 exchange 函数,该函数的功能是实现两个整型变量的值交换。函数原型为

```
void exchange(int *a,int *b);
```

主函数负责输入两个整数,并调用 exchange 函数实现两个整型变量的值交换。要求主函数在调用 exchange 函数前和调用后分别输出两个整型变量的值。

[输入/输出示例]

```
100 200 ↵  
调用前: x=100,y=200  
调用后: x=200,y=100
```

3. 编写程序,对输入的 3 个整数实现从小到大排序并输出。要求通过调用上一题中的 exchange 函数实现必要的变量值交换。

[提示] 可参考如下关键语句:

```
if(a>b) swap(&a,&b);  
if(a>c) swap(&a,&c);  
if(b>c) swap(&b,&c);
```

4. 编写自定义函数,函数原型为 int max_array(int *a,int n);。该函数的功能是接收一个数组的首地址以及数组中元素的个数,并返回该数组中元素的最大值。主函数负责输入 30 个整数,并将其存放在数组中,调用求最大值函数得到数组元素的最大值并输出。

5. 编写自定义函数,函数原型为 int find(int *a,int n,int x);。该函数的功能是在含有 n 个元素的数组 a 中统计元素值等于 x 的个数。要求在主函数中输入 20 个数组元素以及要查找的整数 x,并调用 find 函数来得到并输出数值为 x 的元素个数。

6. 编写自定义函数,函数原型为 int fun(char *p1,char *p2);。该函数的功能是实现两个字符串的比较。设 p1 指向字符串 s1,p2 指向字符串 s2。当 s1 等于 s2 时,函数返回值为 0;当 s1 不等于 s2 时,返回它们二者第一个不相同字符的 ASCII 码差值。两个字符串 s1 和 s2 由 main 函数输入,fun 函数的返回值由 main 函数输出。

7. 编写自定义函数,函数原型为 float fun(float *tt,float *max,float *min);,该函数的功能是计算 tt 数组的最大值、最小值和平均值,平均值通过函数返回值带回,最大值和最小值通过指针参数带回。

[提示] 设参数 tt 指向主函数中的数组 array,参数 max 指向主函数中的变量 max,参数 min 指向主函数中的变量 min。

8. 编写自定义函数,其函数原型为 char *find_replace(char *str,char *str1,char *str2);,其功能是:查找字符串 str 中是否包含子串 str1,如果找到则使用新串 str2 替换旧串 str1。返回值为替换后生成的字符串首地址。主函数负责输入一个长度不超过 80 的字符串,并输入要查找的子串以及替换的子串,然后调用自定义函数进行子串的查找和替换,并输出替换后的新字符串。

[输入/输出示例 1]

```
请输入源字符串:  
com8to8to8cq68to8e ↵  
请输入要查找的子串:
```

8to8 ↵

请输入要替换的子串:

ok ↵

替换后的新串为

comokto8cq6oke

[输入/输出示例 2]

请输入源字符串:

Welcome to Beijing, Beijing is a beautiful city. ↵

请输入要查找的子串:

Beijing ↵

请输入要替换的子串:

Shanghai ↵

替换后的新串为

Welcome to Shanghai, Shanghai is a beautiful city.

9. 针对目前电话用户经常受到垃圾电话骚扰的现状,市场上推出了各种垃圾电话拦截软件。这里尝试实现垃圾电话拦截软件的部分功能,主要实现对来电进行判断,确定该来电是否需要被拦截,我们采取的垃圾电话判定规则是:或者在黑名单中或者以 400 打头。黑名单可以定义为全局的指针数组。编写两个自定义函数,一个自定义函数的功能是判断一个数字串是否以 400 打头,另一个自定义函数的功能是判断一个数字串是否在黑名单中。主函数负责输入一个电话号码,然后分别调用自定义函数对该号码进行判断,并输出该号码是否需要被拦截的信息。

[输入/输出示例 1]

请输入来电号码: 18000112233 ↵

来电 18000112233 不是骚扰电话。

[输入/输出示例 2]

请输入来电号码: 18833339999 ↵

来电 18833339999 是骚扰电话。

[输入/输出示例 3]

请输入来电号码: 4001001234 ↵

来电 4001001234 是骚扰电话。

[提示] “黑名单”是存放了骚扰电话的指针数组。这里可定义如下:

```
char *black[8]={"13010101010","15621212121","18833339999","15277665544",  
               "13422334455","18799887766","13166778899","18912345678"};
```

[说明] 以上电话为随机书写,仅作为示例,如果有和真实电话号码相同的,纯属巧合。