

【教材】《C 语言程序设计（第 2 版）》清华大学出版社，黄保和，江弋 编著。2011 年 10 月第二版。ISBN:978-7-302-26972-4。售价：35 元。

【答案版本】本习题答案为 2012 年 2 月修订版本。

一、选择题

1. 设已定义“int a, *p”，下列赋值表达式中正确的是：C) p = &a
 A. *p = *a B. p = *a C. p = &a D. *p = &a
2. 设已定义“int x, *p=&x;”，则下列表达式中错误的是：B) &*x
 A. *&x B. &*x C. *&p D. &*p
3. 若已定义“int a=1,*b=&a;”，则“printf(“%d\n”,*b);”的输出结果为：A) a 的值。
 A. a 的值 B. a 的地址 C. b 的值 D. b 的地址
4. 设已定义“int x, *p, *p1=&x, *p2=&x;”，则下列表达式中错误的是：C) p=p1+p2.
 A. *p1 + *p2 B. p = p1 C. p = p1 + p2 D. x = p1 - p2
5. 设有函数定义“void p(int *x){printf(“%d\n”,*x);}”和变量定义“int a=3;”，则正确的函数调用是：C) p (&a)
 A. p(a) B. p(*a) C. p(&a) D. p(int *a)
6. 函数“int fun (char * x) {char * y=x; while(*y)y++;return(y-x);}”的功能是A) 求字符串的长度。
 A. 求字符串的长度 B. 比较两个字符串的大小
 C. 将字符串 x 复制到字符串 y D. 将字符串 x 连接到字符串 y 后面
7. 运行一下程序，输出结果为：B) 5 6

```
int fun (int a,int *b)
{
    a++; (*b) ++;
    return a+*b;
}
void main()
{int x=1,y=2;
    printf(“%d”,fun(x,&y));
    printf(“%d”,fun(x,&y));
}
```

 A. 5 5 B. 5 6 C. 6 5 D. 6 6
8. 运行以下程序，输出结果为：C) 58

```
#include<stdio.h>
Int * fun(int a ,int *b)
{a++;(*b)++;
```

```

*b=a+*b;
return b;
}
void main()
{
int x = 1, y = 2, *z;
z = fun(x, &y);
printf("%d", *z);
z = fun(x, &y);
printf("%d", *z);
}

```

A. 5 6 B. 5 7 C. 5 8 D. 6 8

9. 若已定义“int a[]={1,2,3,4},*p=a;”,则下面表达式中值不等于 2 的是 C) *(++a)

A. *(a + 1) B. *(p + 1) C. *(++ a) D. *(++p)

10. 若已定义“int a[]={1,2,3,4},*p=a+1;”,则 p[2]的值为 C) 4

A. 2 B. 3 C. 4 D. 无意义

11. 设已定义“int x[4][10],*p=x[0];”,则下列表达式中的值为整形的是 B) *(p+1)

A. &x[1][2] B. *(p + 1) C. x[1] D. p + 1

12. 设已定义“char s[]="ABCD";”,“printf(“%s”,s+1)”的值为 C) BCD

A. ABCD1 B. B C. BCD D. ABCD

13. 设已定义“char str[]="abcd",*ptr=str;”,则* (prt+4) 的值为 B) 0

A. d B. 0 C. '0' D. 字符 d 的地址

14. 下面对字符串变量的初始化或赋值操作中, 错误的是 C) char a[10];a="OK";

A. char a[] = "OK"; B. char *a = "OK";
C. char a[10] =; a = "OK"; D. char *a; a = "OK";

15. 设已定义“char *ps[2]={"abc"},"1234"};”,则以下叙述中错误的是 A) ps 为指针变量, 它指向一个长度为 2 的字符串数组

A. ps 为指针变量, 它指向一个长度为 2 的字符串数组
B. ps 为指针数组, 其两个元素分别存储字符串 “abc” 和 “1234” 的地址
C. ps[1][2]的值为 3
D. *(ps[0] + 1)

16. 设已定义“struct {int a,b;} s,*ps=&s;”,则错误的结构体成员引用是 C) *ps.a

A. s.a B. ps->a C. *ps.a D. (*ps).a

17. 设已有以下定义, 则表达式的值为 2 的是 A) k=++p->data

```

struct st {

```

```
int data;
st *link;
} a[3]={1,a+1,3,a+2,5,0},*p=a;
```

- A. k = ++p->data B. k = p++->data
C. k = p->data++ D. k = (++p)->data

二、编程题

1. 输入 3 个字符串，输出其中最大的字符串（用字符指针）

```
#include <stdio.h>
#include <stdlib.h>
#define str_count 3
#define str_length 100
int main(int argc, char *argv[])
{
    char a[str_count][str_length],*p;
    printf("请输入 3 个字符串: ");
    int i;
    for(i=0;i<str_count;i++)
    {
        scanf("%s",a[i]);
    }
    p=a[0];
    for(i=1;i<str_count;i++)
    {
        if(strcmp(p,a[i]<0)
        {
            p=a[i];
        }
    }
    printf("最大的字符串为:%s",p);
    system("PAUSE");
    return 0;
}
```

2. 定义一个函数，函数的功能是求已知半径的圆的周长和面积。要求把半径、周长和面积设置成函数参数。

```
#include <stdio.h>
#include <stdlib.h>
#define pi 3.1415926

int main(int argc, char *argv[])
```

```

{
    double r;
    printf("请输入半径: ");
    scanf("%lf",&r);
    double c,s;
    void circle(double r, double *,double *);
    circle(r,&c,&s);
    printf("圆的周长和半径分别为: %lf,%lf",c,s);
    system("PAUSE");
    return 0;
}

void circle(double r,double *c,double *s)
{
    *c=2*pi*r;
    *s=pi*r*r;
}
    
```

3. 定义函数 **max**，函数参数为 3 个字符串，函数返回值最大的字符串。

```

#include <stdio.h>
#include <stdlib.h>
#define str_length 100
#define str_count 3
int main(int argc, char *argv[])
{
    char a[str_count][str_length];
    printf("请输入三个字符串: \n");
    int i;
    for(i=0;i<str_count;i++)
    {
        scanf("%s",a[i]);
    }
    char * max(char * a1,char *a2,char *a3);
    printf("最大的字符串是: %s",max(a[0],a[1],a[2]));
    system("PAUSE");
    return 0;
}

char * max(char * a1,char *a2,char *a3)
{
    char * p;
    p=a1;
    
```

```

if(strcmp(p,a2)<0)
{
    p=a2;
}
if(strcmp(p,a3)<0)
{
    p=a3;
}
return p;
}
    
```

4. 自己定义字符串复制函数 strcpy，然后调用该函数

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char *s="test";
    char *o;
    char * str_copy_self(char *s);
    o=str_copy_self(s);
    printf("%s",o);
    system("PAUSE");
    return 0;
}

char * str_copy_self(char *s)
{
    int str_length=strlen(s);
    char *o=(char *)malloc((str_length+1)*sizeof(char));
    int i;
    for(i=0;i<str_length;i++)
    {
        o[i]=s[i];
    }
    o[i]='\0';
    return o;
}
    
```

5. 定义一个函数，函数参数为一维数组（用指针表示），函数返回数组元素的平均值。

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    double arr[]={ 1,2,3,4,5,6,7,8,9,10};
    double ave(double arr[],int arr_len);
    printf("%lf",ave(arr,10));
    system("PAUSE");
    return 0;
}

double ave(double arr[],int arr_len)
{
    int i;
    double res=0;
    for(i=0;i<arr_len;i++)
    {
        res+=arr[i];
    }
    res=res/arr_len;
    return res;
}
    
```

6. 定义一个函数，删除字符串中第 k 个字符开始的 m 个字符，例如删除字符串 `abcde` 第 2 个字符开始的 3 个字符，则删除后结果为 `ae`；又如删除字符串 `abcde` 第 4 个字符开始的 5 个字符，则删除后结果为 `abc`；

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char str[]="abcde";
    void str_pruning(char * str,int begin_pos,int pruning_count);
    str_pruning(str,4,5);
    printf("%s",str);
    system("PAUSE");
    return 0;
}

void str_pruning(char * str,int begin_pos,int pruning_count)
    
```

```

{
    int str_len=strlen(str);

    if(begin_pos>=str_len)
    {
        str[0]='\0';
    }
    else if((begin_pos-1+pruning_count)>=str_len)
    {
        str[begin_pos-1]='\0';
    }
    else
    {
        int i=0;
        while(1)
        {

            str[begin_pos+i-1]=str[begin_pos+pruning_count+i-1];
            if(str[begin_pos+pruning_count+i-1]=='\0')
            {
                break;
            }
            i++;
        }

    }

}
    
```

7. 在字符串中删除所有指定字符（如把字符串 `teacher` 中的 `e` 字符删除，得到 `tachr`），使用子函数和字符指针。

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    char str[]="teacher";
    void char_punching(char * str, char l);
    char_punching(str,'e');
    printf("%s",str);
    system("PAUSE");
    return 0;
}
    
```

```

    }

void char_punching(char * str, char l)
{
    int c=0;
    int str_len=strlen(str);
    int i;
    for(i=0;i<str_len;i++)
    {
        if(str[i]==l)
        {
            c++;
        }
        else
        {
            str[i-c]=str[i];
        }
    }
    str[str_len-c]='\0';
}
    
```

8. 求二维数组的最大元素值及最大元素的位置（用指针法引用数组元素）

```

#include <stdlib.h>

int main(int argc, char *argv[])
{
    int a[3][3]={{1,2,333},{4,15,6},{7,8,9}};
    int x=3,y=3,i,j;
    int px=0,py=0;
    int *p,*p_max;
    p_max=a[0];
    for(i=0;i<x;i++)
    {
        p=a[i];
        for(j=0;j<y;j++)
        {
            printf("%d\n",*p);
            if(*p_max<*p)
            {
                px=i;
                py=j;
                p_max=p;
            }
        }
    }
}
    
```



```
        }
        p++;
    }
}
printf("%d\n",*p_max);
system("PAUSE");
return 0;
}
```

=====此后无内容=====