

1.设有说明 **int s[2]={0,1},*p=s;**则下列错误的 C 语句是 () 。

A) s+=1; B) p+=1; C) *p++; D) (*p)++;

2.**int array[10] = {1,2,3,4,5,6,7,8,9,10};**

int *p = NULL; p = array;

**p+1; *(p + 1); (p + 10); *(p + 10); p = array + 2; p
-; &(p = p+3);**

各个语句的值。

3.下面程序的输出:

#include <stdio.h>

fun(int x, int y, int *cp, int *dp)

{

***cp=x+y;**

***dp=x-y;**

}

int main()

{

int a,b,c,d;

a=30; b=50;

fun(a, b, &c, &d);

printf("%d,%d\n", c, d);

return 0;

}

A) 50,30 B) 30,50 C) 80,-20 D) 80,20

4.若有定义和语句:

int **pp,*p,a=10,b=20;

pp=&p; p=&a; p=&b;

printf("%d,%d\n",*p,pp);**

则输出结果是 ()

- A) 10,20 B) 10,10 C) 20,10 D) 20,20**

5.下面程序的运行结果:

```
char s[80], *sp = "HELLO!"  
sp = strcpy(s,sp);  
s[0] = 'h';  
puts(sp);
```

6.程序中输入 **apple<CR> cat <CR>**, 输出结果是

```
main ()  
{  
    char *s , c[80];  
    s = c;  
    gets(s);  
    while ((*++s)!='\0')  
    {  
        if (*s == 'a')  
            break;  
        else  
        {  
            s++;  
            gets(s);  
        }  
    }  
    puts(s);  
}
```

7.**mystrlen** 函数的功能是计算 **str** 所指字符串的长度,并作为函数值返回。请填空。

```

int mystrlen(char *str)
{
    int i;
    for(i=0; ____!='\0';i++);
    return(____);
}

```

8.有如下程序段

```

int *p,a=10,b=1;
p=&a; a=*p+b;

```

执行该程序段后，**a** 的值为

9.若有以下定义和语句：

```

double r=99, *p=&r;
*p=r;

```

则以下正确的叙述是

- A)** 以下两处的***p** 含义相同，都说明给指针变量 **p** 赋值
- B)** 在"**double r=99, *p=&r;**"中，把 **r** 的地址赋值给了 **p** 所指的存储单元
- C)** 语句"***p=r;**"把变量 **r** 的值赋给指针变量 **p**
- D)** 语句"***p=r;**"取变量 **r** 的值放回 **r** 中

10.下列程序的输出结果是

```

#include
main( )
{
    char b[30], *chp;
    strcpy(&b[0], "CH");
    strcpy(&b[1], "DEF");
    strcpy(&b[2], "ABC");
}

```

```
    printf("%s \n", b);  
}
```

11. 设 **p1** 和 **p2** 是指向同一个 **int** 型一维数组的指针变量, **k** 为 **int** 型变量, 则不能正确执行的语句是

A) k=*p1+*p2; B) p2=k; C) p1=p2; D) k=*p1 * (*p2);

12. 若有以下定义:

```
char s[20]="programming", *ps=s;
```

则不能代表字符 **o** 的表达式是

**A) ps+2 B) s[2] C) ps[2]
D) ps+=2, *ps**

13. 若有以下的定义:

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

则值为 **3** 的表式是

**A) p+=2, *(p++) B) p+=2, *++p C) p+=3, *p++ D)
p+=2, ++*p**

14. 若有以下说明:

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

则数值为 **6** 的表达式是

A) *p+6 B) *(p+6) C) *p+=5 D) p+5

15. 下面程序的输出结果是_____。

```
int main()  
{  
    int a[]={1,2,3,4,5,6};
```

```

int *p;
p=a ;
printf("%d ",*p);
printf("%d ",*(++p));
printf("%d ",*++p);
printf("%d ",*(p--)) ;
p+=3;
printf("%d %d ",*p,*(a+3));
}

```

A) 1 2 3 3 5 4 B) 1 2 3 4 5 6 C) 1 2 2 3 4 5 D) 1 2 3 4 4 5

16.以下程序的输出结果是

```

int main()
{
    char *p="abcdefgh",*r;
    long *q;
    q=(long*)p;
    q++;
    r=(char*)q;
    printf("%s\n",r);
}

```

17.有如下说明

```
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

18.函数 **mycmp (char*s,chat *t)**的功能是比较字符串 **s** 和 **t** 的大小，当 **s** 等于 **t** 时返回 **0**，否则返回 **s** 和 **t** 的第一个不同字符的 **ASCII** 码的差值，即当 **s>t** 时返回正值，当 **s<t** 时返回负值，请填空。

```
mycmp(char *s, char *t)
{
    while(*s==*t)
    {
        if(_____)
            return 0;
        ++s; ++t;
    }
    return ();
}
```

19.下列程序的输出结果是

```
int main( )
{
    char*p1,*p2,str[50]="xyz";
    Char str2[50] = "abcd"; "abcdBCD"
    p1=str2;
    p2="ABCD";
    strcpy(str+2, strcat(p1+2,p2+1));
    printf("%s",str);
}
```

A) xyabcAB B) abcABz C) ABabcz D)
xycdBCD

20.已定义以下函数

```
fun(char *p2, char *p1)
{
```

```

while((*p2=*p1)!='\0')
{
    p1++;
    p2++;
}
}

```

函数的功能是

- A)** 将 **p1** 所指字符串复制到 **p2** 所指内存空间
- B)** 将 **p1** 所指字符串的地址赋给指针 **p2**
- C)** 对 **p1** 和 **p2** 两个指针所指字符串进行比较
- D)** 检查 **p1** 和 **p2** 两个指针所指字符串中是否有 **'\0'**

21.以下程序的输出结果是

```

main( )
{
    char *p1, *p2, str[50]="ABCDEFGFG";
    p1="abcd";
    p2="efgh";
    strcpy(str+1, p2+1);
    strcpy(str+3, p1+3);
    printf("%s", str);
}

```


22.下面语句初始化后，执行 **p+1** 的结果

```

int s[2][3] = {0}, (*p)[3];
p = s;

```

23.执行以下程序段后,m 的值为

```
int a[2][3]={ {1,2,3},{4,5,6} };  
int m,*p;  
p=&a[0][0];  
m = (*p)*(*(p+2))*(*(p+4));
```

A) 15

B) 14

C) 13

D) 12

24.有以下程序

```
#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);  
}
```

若从键盘输入: **abc def**<回车>, 则输出结果是:

25.若有说明: **long *p, a;** 则不能通过 **scanf** 语句正确给输入项读入数据的程序段是

- A)*p=&a; scanf("%ld", p);
- B)p=(long *)malloc(8); scanf("%ld", p);
- C)scanf("%ld", p=&a);
- D)scanf("%ld", &a);

26.若有以下说明和定义

```
int fun(int *c){  };
```



```

main()
{
    int (*a)(int*)=fun,*b(),w[10],c;
    □
}

```

在必要的赋值之后，对 **fun** 函数的正确调用语句是

A) a=a(w); B) (*a)(&c); C) b=*b(w); D) fun (b);

27.若已定义 **char s[2][40]={“china”,“English”}**；则在下面表达式中不能表示'E'字符的是（ ）。

A) s[1][0] B) *(s+1) C) **(s+1) D) *s[1]

28.若有以下说明和语句，对 **C** 数组元素的正确引用是（ ）。

```

int c[4][5],(*cp)[5];    cp=c;
A) cp+1            B) *(cp+3)      C) *(cp+1)+3
D) *(*cp+2)

```

29.定义 **int (* p)()** 指针可以

- A** 代表函数的返回值
- B** 指向函数的入口地址
- C** 代表函数的类型
- D** 表示函数返回值的类型

30.若有以下定义和语句：

```

int w[2][3],(*pw)[3];
pw=w;

```

则对 **w** 数组元素非法引用是

A)*(w[0]+2)

B)*(pw+1)[2]

C)pw[0][0]

D)*(pw[1]+2)