

## C 语言指针练习题

姓名:\_\_\_\_\_

分数 (总分 100) :\_\_\_\_\_

选择题答题处(请答完题后把答案写到答题卡)(37\*2=74):

1~10										
11~20										
21~30										
31~37										

### 一、填空题 (每空 2 分 13\*2 分)

- (1) 若有以下定义, 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\_\_\_\_\_。

int w[10]={23, 54, 10, 33, 47, 98, 72, 80, 61}, \*p=w;

- (2) 在说明语句: int \*f(); 中, 标识符 f 代表的是\_\_\_\_\_

- (3) 以下程序通过函数指针 p 调用函数 fun, 请在填空栏内, 写出定义变量 p 的语句。

```
void fun(int *x, int *y)
{ ..... }
int main()
{ int a=10, b=20;
  _____; /*定义变量 p */
  p=fun; p(&a, &b);
  .....
}
```

- (4) 下列程序的输出结果是\_\_\_\_\_。

```
void fun(int *n)
{ while( (*n)--);
  printf( "%d", ++(*n));
}
int main()
{ int a=100;
  fun(&a);
}
```

- (5) 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);}
```

- (6) 以下程序中, select 函数的功能是: 在 N 行 M 列的二维数组中, 选出一个最大值作为函数值返回, 并通过形参传回此最大值所在的行下标。请填空。

```
#define N 3
#define M 3
int select(int a[N][M], int *n)
{ int i, j, row=1, colum=1;
  for(i=0; i<N; i++)
    for(j=0; j<M; j++)
      if(a[i][j]>a[row][colum])
        {row=i; colum=j;}
  *n= _____;
  return _____;
}
int main()
{ int a[N][M]={9, 11, 23, 6, 1, 15, 9, 17, 20}, max, n;
  max=select(a, &n);
  printf("max=%d, line=%d\n", max, n) ;
}
```

- (7) 设有以下程序:

```
#include <stdio.h>
main()
{
  int a, b, k=4, m=6, *p1=&k, *p2=&m;
  a=p1==&m;
  b=(*p1)/(*p2)+7;
  printf("a=%d\n", a);
  printf("b=%d\n", b) ;
}
```

执行该程序后, a 的值为\_\_\_\_\_, b 的值为\_\_\_\_\_。

(8) 以下程序运行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>

int main()
{
    char s[]="9876",*p;
    for ( p=s ; *p>'7' ;p++)
        printf("%s\n", p);
}
```

(9) 以下程序

```
int main()
{
    char *p = "abcdef" ;
    printf( "%c ",*(p+2));
    p +=2;
    *p = 'm' ;
    Printf( "%c\n", *p);
}
```

执行后的结果是 \_\_\_\_\_

(10) 以下程序的输出结果是\_\_\_\_\_

```
int main()
{
    char s[] = "hello\0world" ;
    char *p = s + 2;
    printf( "%s ",p);
    printf( "%s\n",p+4);
}
```

(11) 以下程序运行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
#include <string.h>
main( )
{ char ch[ ] =" abc", x[3][4]; int i;
  for(i=0;i<3;i++) strcpy(x[i], ch );
  for(i=0;i<3;i++) printf( "%s", &x[i][i] );
  printf("\n"); }
```

## 二、选择题 (38\*2 分)

(1) 若有说明: int i, j=7,\*p=&i;, 则与 i=j;等价的语句是 \_\_\_\_\_

- A) i=\*p;                      B) \*p=&\*j;  
C) i=&j;                      D) i==\*p;

(2) 以下程序执行后 a 的值是\_\_\_\_\_

```
int main()
{
    int a,k=4,m=6,*p1=&k,*p2=&m;
    a=p1==&m;
```

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

- A) 4    B) 1    C) 0    D) 运行时出错, a 无定值

(3) 以下程序运行后, 输出结果是 \_\_\_\_\_

```
int main()
{char *s= "abcde";
  s=s+2;
  printf( "%ld \n", s); }
```

- A. cde    B. 字符 c 的 ASCLL 码值    C. 字符 C 的地址    D. 出错

(4) 下列程序的输出结果是\_\_\_\_\_

```
main()
{int a[5]={2,4,6,8,10}, *p,**k;
  p=a;
  k=&p;
  printf( "%d", *(p));
  printf( "%d\n", **k);
}
```

- A) 44    B) 22    C) 24    D) 46

(5) . 若有下面的程序片段

```
int a[12]={0}, *p[3], **pp,i;
for(i=0;i<3;i++)
  p[i]=&a[i*4];
pp=p;
```

则对数组元素的错误引用是\_\_\_\_\_

- A、 pp[0][1]    B、 a[10]    C、 p[3][1]    D、 \*((p+2) + 2)

(6) 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

(7) 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

- A) 12    B) 11    C) 10    D) 编译出错

(8) 对于基类型相同的两个指针变量之间, 不能进行的运算是

- A) <    B) =    C) +    D) -

(9) 有以下程序\_\_\_\_\_

```
main()
{ char str[]="xyz",*ps=str;
  while(*ps) ps++;
  for(ps--;ps-str>=0;ps--)
    puts(ps); }
```

}

执行后输出结果是

- A) yz<回车>xyz                      B) z<回车>yzy  
C) z<回车>yzy<回车>xyz<回车>  
D) x<回车>xy<回车>xyz

(10) 若定义了以下函数: \_\_\_\_\_

```
void f(.....)
{.....
 *p=(double *)malloc(10*sizeof( double));
.....
}
```

p 是该函数的形参, 要求通过 p 把动态分配存储单元的地址传回主调函数, 则形参 p 的正确定义应当是

- A) double \*p B) float \*\*p C) double \*\*p D) float \*p

(11) 下列程序的输出结果是\_\_\_\_\_

```
int main()
{ char a[10]={9,8,7,6,5,4,3,2,1,0}, *p=a+5;
 printf( "%d", *--p);
}
```

- A) 非法      B) a[4]的地址      C) 5      D) 3

(12) 下列程序的运行结果是\_\_\_\_\_

```
void fun(int *a, int *b)
{ int *k;
 k=a; a=b; b=k;}
int main()
{ int a=3, b=6, *x=&a, *y=&b;
 fun(x, y);
 printf( "%d %d", a, b);
}
```

- A) 6 3                      B) 3 6  
C) 编译出错              D) 0 0

(13) 下面程序的输出结果是\_\_\_\_\_

```
int main()
{ int a[ ]={1,2,3,4,5,6,7,8,9, 0}, *p;
 p=a;
 printf( "%d\n", *p+9);      }
```

- A) 0    B) 1    C) 10    D) 9

(14) 若定义: int a=511, \*b=&a;, 则 printf("%d\n", \*b); 的输出结果为 \_\_\_\_\_

- A) 无确定值      B) a 的地址      C) 512      D) 511

(15) 以下程序的输出结果是 \_\_\_\_\_

```
char cchar(char ch)
{ if(ch>= 'A' &&ch<= 'Z') ch=ch- 'A'+ 'a';
 return ch;
```

}

int main()

```
{ char s[]="ABC+abc=defDEF", * p=s;
 while(*p)
 { *p=cchar(*p);
 p++;
 }
 printf("%s\n", s);
}
```

- A) abc+ABC=DEFdef B) abc+abc=defdef  
C) abcaABCDEFdef D) abcabcdefdef

(16) 若指针 p 已正确定义, 要使 p 指向两个连续的整型动态存储单元, 不正确

的语句是 \_\_\_\_\_

- A) p=2\*(int\*)malloc(sizeof(int));  
B) p=(int\*)malloc(2\*sizeof(int)) ;  
C) p=(int\*)malloc(2\*2);  
D) p=(int\*)calloc(2, sizeof(int)) ;

(17) 若有以下定义和语句: \_\_\_\_\_

```
int s[4][5], (*ps)[5];
ps=s;
```

则对 s 数组元素的正确引用形式是

- A) ps+1 B) \*(ps +3) C) ps[0][2] D) \*(ps+1)+3

(18) 以下程序的输出结果是 \_\_\_\_\_

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

- A) 8    B) 3    C) 1    D) 7

(19) 若有说明: int i, j=2, \*p=&i;, 则能完成 i=j 赋值功能的语句是\_\_\_\_\_

- A) i=\*p; B) \*p=\*&j; C) i= &j; D) i=\*\*p;

(20) 若有说明: int n=2, \*p=&n, \*q=p;, 则以下非法的赋值语句是\_\_\_\_\_

- A) p=q; B) \*p=\*q; C) n=\*q; D) p=n;

(21) 有以下程序 \_\_\_\_\_

```
void fun(char *c, int d)
{ *c=*c+1;
 d=d+1;
```

```

    printf( "%c,%c, ", *c, d);
}
int main()
{ char a=' A' ,b=' a' ;
  fun(&b,a);
  printf( "%c,%c\n", a ,b);
}

```

程序运行后的输出结果是

- A) B, a, B, a    B) a, B, a, B    C) A, b, A, b    D) b, B, A, b

(22) 有以下程序 \_\_\_\_\_

```

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

```

程序运行后的输出结果是

- A) 12    B) 15    C) 6    D) 5

(23) 有以下程序 \_\_\_\_\_

```

void ss(char *s, char t)
{
    while(*s)
    {if(*s==t)
        *s=t- 'a' +' A' ;
        s++;
    }
}
int main()
{char str1[100]= "abcddfefdbd" , c=' d' ;
  ss(str1, c);
  printf( "%s\n", str1);
}

```

程序运行后的输出结果是

- A) ABCDDEFEDBD    B) abcDDfefDbD  
C) abcAAfefAbA    D) Abcddfefdbd

(24) 有以下程序 \_\_\_\_\_

```

#include "stdio.h"
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: abcde f<回车>, 则输出结果是:

- A) defdef    B) abcdef    C) abcd    D) ff

(25) 下面程序的运行结果是: \_\_\_\_\_

```

void swap(int*a, int*b)
{int *t;
  t=a;a=b;b=t;
}
int main()
{ int x=3,y=5,*p=&x,*q=&y;
  swap(p,q);
  printf( "%d%d\n", *p,*q);
}

```

- A) 33    B) 35    C) 53    D) 55

(26) 若有一些定义和语句 \_\_\_\_\_

```

#include <stdio.h>
int a=4,b=3,*p,*q,*w;
p=&a; q=&b; w=q; q=NULL;
则以下选项中错误的语句是
A) *q=0; B) w=p; C) *p=a; D) *p=*w;

```

(27) 有以下程序 \_\_\_\_\_

```

int *f(int *x, int *y)
{ if(*x<*y)
    return x;
  else return y;
}
main()
{ int a=7,b=8,*p,*q,*r;
  p=&a; q=&b;
  r=f(p,q);
  printf("%d,%d,%d\n", *p,*q,*r);
}

```

执行后输出结果是

- A) 7, 8, 8    B) 7, 8, 7    C) 8, 7, 7    D) 8, 7, 8

(28) 有以下程序 \_\_\_\_\_

```

main()
{ char *s[]={"one","two","three"},*p;
  p=s[1];
  printf("%c,%s\n",*(p+1),s[0]);
}

```

执行后输出结果是

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

(29) 有以下程序 \_\_\_\_\_

```

main()
{ int x[8]={8,7,6,5,0,0},*s;
  s=x+3;
  printf("%d\n",s[2]);
}

```

执行后输出结果是

A) 随机值 B) 0 C) 5 D) 6

(30) 以下函数的功能是：通过键盘输入数据，为数组中的所有元素赋值。

```
#define N 10
void arrin(int x[N])
{ int i=0;
  while(i<N)
    scanf("%d",_____);
}
```

在下划线处应填入的是\_\_\_\_\_

A) x+i B) &x[i+1] C) x+(i++) D) &x[i++]

(31) 有以下程序

```
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]);
}
```

执行后输出结果是 \_\_\_\_\_

A) 7 B) 6 C) 8 D) 无确定值

(32) 有以下程序

```
void fun(int *a,int i,int j)
{ int t;
  if (i<j)
  { t=a[i];a[i]=a[j];a[j]=t;
    fun(a,++i,--j);
  }
}

int main()
{ int a[]={1,2,3,4,5,6}, i;
  fun(a,0,5);
  for(i=0;i<6;i++)
    printf("%d",a[i]);
}
```

执行后输出结果是 \_\_\_\_\_

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

(33) 有以下程序

```
int main()
{
  char buf[] = "abcdefg";
  char *p;
  *p = buf[1];
}
```

```
printf( "%c %c\n", *p, *(p+3));
}
```

执行后输出结果是 \_\_\_\_\_

A) a d B) b b C) b e D) 不确定

(34) #include <string.h>

```
main( )
{ char p[20]={ 'a','b','c','d' };
  char q[ ]="abc",r[ ]="abcde";
  strcpy(p+strlen(q),r);
  strcat(p,q);
  printf("%d %d\n",sizeof(p),strlen(p));
}
```

程序运行后的输出结果是：

A) 20 9 B) 9 9  
C) 20 11 D) 11 11

(35) 以下程序运行后的输出结果是： \_\_\_\_\_

```
#include <string.h>
int main( )
{ char p[20]={ 'a','b','c','d' };
  char q[ ]="abc",r[ ]="abcde";
  strcat(p,r);
  strcpy(p+strlen(q),q);
  printf("%d\n",strlen(p));
}
```

A) 9 B) 6 C) 11 D) 7

(36) 若有定义语句：char s[10]="1234567\0\0";，则 strlen(s)的值是\_\_\_\_\_。

A) 7 B) 8 C) 9 D) 10

(37) 以下关于字符串的叙述中正确的是\_\_\_\_\_。

- A) C 语言中有字符串类型的常量和变量
- B) 两个字符串中的字符个数相同时才能进行字符串大小的比较
- C) 可以用关系运算符对字符串的大小进行比较
- D) 空串一定比空格打头的字符串小