1． 若有说明：int a=2, \*p=&a, \*q=p;，则以下非法的赋值语句是（）。

A． p=q;           B． \*p=\*q;        C． a=\*q;         D． q=a;

2． 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

3． 已有定义　int a=2, \*p1=&a, \*p2=&a; 下面不能正确执行的赋值语句是（）。

A． a=\*p1+\*p2;     B． p1=a;          C． p1=p2;        D． a=\*p1\*(\*p2);

4．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

5． 若有说明语句：int a, b, c, \*d=&c；，则能正确从键盘读入三个整数分别赋给变量a、b、c的语句是（）。

A． scanf("%d%d%d", &a, &b, d);

B． scanf("%d%d%d", a, b, d);

C． scanf("%d%d%d", &a, &b, &d);

D． scanf("%d%d%d", a, b,\*d);

6．若已定义int a=5; 下面对（1）、（2）两个语句的正确解释是（）。

(1) int \*p=&a;                 (2) \*p=a;

A． 语句（1）和（2）中的\*p含义相同，都表示给指针变量p赋值。

B． (1)和(2)语句的执行结果，都是把变量a的地址值赋给指针变量p。

C． (1)在对p进行说明的同时进行初始化，使p指向a；

(2)变量a的值赋给指针变量p。

D． (1)在对p进行说明的同时进行初始化，使p指向a；

  (2)将变量a的值赋予\*p。

7．若有语句int \*p, a=10; p=&a; 下面均代表地址的一组选项是（）。

A． a, p, \*&a                             B． &\*a, &a, \*p

C． \*&p, \*p,&a                           D．&a, &\*p, p

8． 若需要建立如图所示的存储结构，且已有说明double \*p, x=0.2345; 则正确的赋值语句是（）。

A． p=x;                                  B． p=&x;

C． \*p=x;                                 D． \*p=&x;

9．若有说明：int \*p, a=1, b; 以下正确的程序段是（）。

A．p=&b;                                 B． scanf("%d",&b);

scanf("%d", &p);                            \*p=b;

C． p=&b;                                 D． p=&b;

scanf("%d", \*p);                             \*p=a;

10．有如下语句：int m=6, n=9, \*p, \*q; p=&m; q=&n; 如图1所示，若要实现下图所示的存储结构，可选用的赋值语句是（）。

    A． \*p=\*q;                                B． p=\*q;

    C． p=q;                                  D． \*p=q;

11．以下程序中调用scanf函数给变量a输入数值的方法是错误的，其错误原因是（）。

   #include <stdio.h>

   main()

{

int \*p, \*q, a, b;

p=&a;

printf("input a:");

scanf("%d", \*p);

…

}

A． \*p表示的是指针变量p的地址

B． \*p表示的是变量a的值，而不是变量a的地址

C． \*p表示的是指针变量p的值

D． \*p只能用来说明p是一个指针变量

12．下面程序的功能是从输入的十个字符串中找出最长的那个串。请在\_\_\_\_\_\_\_\_处填空。

   #include "stdio.h"

   #include "string.h"

   #define N 10

   main()

{

char s[N][81], \* t;

int j;

  for (j=0; j<N; j++)

 gets (s[j]);

t= \*s;

  for (j=1; j<N; j++)

 if(strlen(t)<strlen(s[j]))   \_\_\_\_\_\_\_\_;

   printf("the max length of ten strings is: %d, %s\n", strlen(t), t);

}

A． t=s[j]      B． t=&s[j]         C． t= s++          D． t=s[j][0]

13．下面判断正确的是（）。

A． char \*s="girl";                等价于 char \*s; \*s="girl";

B． char s[10]={"girl"};            等价于  char s[10]; s[10]={"girl"};

C． char \*s="girl";                等价于   char \*s; s="girl";

D． char s[4]= "boy", t[4]= "boy";   等价于　char s[4]=t[4]= "boy"

14．设char \*s="\ta\017bc"；则指针变量s指向的字符串所占的字节数是（）。

A． 9                   B． 5            C．  6             D． 7

15．下面程序段中，for循环的执行次数是（）。

   char \*s="\ta\018bc";

   for (；\*s!='\0'；s++) printf("\*");

A． 9                   B． 5            C．  6             D．7

16．以下不能正确进行字符串赋初值的语句是（）。

A.     char str[5]= "good!";

B． char \*str="good!";

C.     char str[]="good!";

    D． char str[5]={'g', 'o','o', 'd'};

17．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));

18．下面程序段的运行结果是（）。

　　char \*s="abcde";

   s+=2;

   printf("%d", s);

A． cde      B．字符'c'    C．字符'c'的地址     D．无确定的输出结果

19．设有如下的程序段：char s[]="girl", \*t;   t=s;则下列叙述正确的是（）。

A． s和t完全相同

B． 数组s中的内容和指针变量t中的内容相等

C． s数组长度和t所指向的字符串长度相等

D． \*t与s[0]相等

20．以下正确的程序段是（）。

A． char s[20];                               B． char \*s;

  scanf("%s", &s);                             scanf("%s", s);

C． char s[20];                              D． char s[20], \*t=s;

    scanf("%s",&s[2]);                           scanf("%s", t[2]);

21．下面程序段的运行结果是（）。

#include "stdio.h"

main()

{

 int m=10, n=20;

  char \*format="%s, m=%d, n=%d\n";

  m\*=n;

 printf(format, "m\*=n", m,n);

}

A． format, "m\*=n", m, n                      B． format, "m\*=n"

C． m\*=n, m=200, n=20                        D． 以上结果都不对

22．以下与库函数strcpy(char \*p, char \*q)功能不相等的程序段是（）。

A． strcpy1(char \*p, char \*q)

{ while ((\*p++=\*q++)!='\0');

}

B． strcpy2( char \*p, char \*q)

{   while((\*p=\*q)!='\0')

        {p++;      q++;}

}

C． strcpy3(char\*p, char \*q)

{  while (\*p++=\*q++);

}

   D． strcpy4(char \*p, char \*q)

      {   while(\*p)

             \*p++=\*q++;

}

23．下面程序段的运行结果是（）。

#include "stdio.h"

   main()

{

char s[]="example!", \*t;

      t=s;

       while( \*t!='p')

{ printf("%c",\*t-32);

t++;}

}

    A．EXAMPLE!         B． example!         C． EXAM         D． example!

24．以下正确的程序段是（）。

A． char s[]="12345", t[]="6543d21";    strcpy( s,t);

B． char s[20],\*t="12345";   strcat(s,t);

C． char s[20]=" ", \*t="12345";strcat(s, t);

D． char \*s="12345", \*t="54321";  strcat (s,t);

25．以下与库函数strcmp(char \*s, chat \*t)的功能相等的程序段是（）。

A． strcmp1(char \*s, chat \*t)

{  for ( ；\*s++==\*t++ ；)

        if (\*s=='\0') return 0;

    return (\*s-\*t);

}

B． strcmp2(char \*s, char \*t)

  {  for( ；\*s++==\*t++ ；)

if (!\*s) return0;

         return  (\*s-\*t);

  C． strcmp3(char \*s, char \*t)

    {  for ( ；\*t==\*s ；)

{  if (!\*t)  return 0;

   t++;

   s++;}

           return (\*s-\*t);

 }

D． strcmp4(char \*s, char \*t)

{   for( ；\*s==\*t；s++,t++)

    if (!\*s) return 0;

return (\*t-\*s);

}

26．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

27．不合法的main函数命令行参数表示形式是：（）。

A． main( int a,char \*c[])                  B． main(int argc, char \*argv)

C． main( intarc, char \*\*arv)               D． main( int argv, char\*argc[])

28．若有说明语句：char  s[]="it is a example．", \*t="it is aexample.";则以下不正确的叙述（）。

A． s表示的是第一个字符i的地址，s+1表示的是第二个字符t的地址

B． t指向另外的字符串时，字符串的长度不受限制

C． t变量中存放的地址值可以改变

D． s中只能存放16个字符

29．若已定义char s[10];则在下面表达式中不表示s[1]地址的是（）。

A． s+1          B． s++           C． &s[0]+1        D． &s[1]

30．下面程序段的运行结果是（）。（注：└┘代表空格）

   #include "stdio.h"

   main()

   {   char s[6];

s="abcd";

printf("\"%s\"\n", s);

}

    A． "abcd"       B． "abcd└┘"       C． \"abcd\"      D．编译出错

31．执行以下程序后，a的值为【1】， b的值为【2】。

#include <stdio.h>

main()

{

int a, b, k=4, m=6, \*p=&k, \*q=&m;

    a=p==&m;

b=(-\*p)/(\*q)+7;

printf("a=%d\n", a);

printf("b=%d\n", b);

}

【1】 A． -1            B． 1            C． 0             D． 4

【2】 A． 5             B． 6            C． 7              D． 10

32．下面程序的功能是将字符串s的所有字符传送到字符串t中，要求每传递三个字符后再存放一个空格，例如字符串s为"abcdefg"，则字符串t为"abc def g",请选择填空。

#include "stdio.h"

   #include "string.h"

   main()

{

int j, k=0;

char s[60], t[100], \*p;

p=s;

gets(p);

while(\*p)

{  for (j=1; j<=3&& \*p; 【1】) t[k]=\*p;

if (【2】) { t[k]=' ';k++;}

}

t[k]='\0';

puts(t);

}

    【1】 A．p++        B． p++,k++        C． p++, k++, j++       D． k++, j++

【2】 A． j==4       B． \*p=='\0'       C． !\*p                D． j!=4

33．下面程序的功能是将八进制正整数字符串转换为十进制整数。请选择填空。

#include "stdio.h"

#include "string.h"

main()

{

char \*t, s[8];

  int n;

  t=s;

  gets(t);

  n=【1】;

  while (【2】!= '\0') n=n\*8+\*t-'0';

 printf("%d\n",n);

}

【1】A． 0           B． \*t              C． \*t-'0'             D． \*t+'0'

【2】A．\*t          B． \*t++            C． \*(++t)              D． t

34．下面程序的功能是在字符串s中找出最大的字符并放在第一个位置上，并将该字符前的原字符往后顺序移动，如：boy&girl变成ybo&girl。请选择填空。

#include "stdio.h"

#include "string.h"

main()

{

char s[80], \*t, max, \*w;

   t=s;

   gets(t);

   max=\*(t++);

   while (\*t!='\0')

     {

if (max<\*t)

 { max=\*t; w=t; }

          t++;

      }

    t=w;

    while (【1】)

      {

\*t=\*(t-1);

         【2】;}

    \*t=max;

    puts(t);

 }

【1】A．t>s          B． t>=s             C． \*t>s[0]          D． \*t>=s[0]

【2】A． t++          B． s--              C． t--              D． w--

35．以下程序的功能是删除字符串s中的所有空格（包括TAB符、回车符），请填空。

   #include "stdio.h"

   #include "string.h"

   #include "ctype.h"

   main()

{

char  s[80];

gets(s);

delspace(s);

puts(s);

}

    delspace(char \*t)

{

int m, n;

char c[80];

for(m=0, n=0; 【1】;m++)

  if (!isspace(【2】)) /\*C语言提供的库函数，用以判断字符是否为空格\*/

{

 c[n]=t[m];

  n++;

}

c[n]='\0';

strcpy(t, c);

}

    【1】A． t[m]         B． !t[m]           C． t[m]='\0'       D． t[m]=='\0'

【2】A． t+m         B． \*c[m]           C． \*(t+m)         D．\*(c+m)

36．下面程序的功能是统计字串sub在母串s中出现的次数。请选择填空。

   #include "stdio.h"

   #include "string.h"

   main()

{

char s[80], sub[80];

int n;

gets(s);

gets(sub);

printf("%d\n", count(s,sub));

}

int count( char \*p, char \*q)

{

int m, n, k, num=0;

 for (m=0; p[m]; m++)

     for (【1】,k=0; q[k]==p[n]; k++, n++)

          if(q[【2】]=='\0')

            { num++;  break;}

  return (num);

}

【1】A． n=m+1       B．n=m            C． n=0              D． n=1

【2】A． k            B． k++            C． k+1              D． ++k

37．下列程序的输出结果是（）。

#include "stdio.h"

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

38．以下程序的输出结果是（）。

#include "stdio.h"

char cchar(char ch)

{

 if (ch>='A' && ch<='Z')  ch=ch-'A'+'a';

 return ch;

}

main()

{

char s[]="ABC+abc=defDEF", \*p=s;

while(\*p)

{

\*p=cchar(\*p);

    p++;

}

printf("%s\n",s);

}

A． abc+ABC=DEFdef                          B． abcaABCDEFdef

C． abc+abc=defdef                              D． abcabcdefdef

39．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);

     printf("%d\n", strlen(b2));

    }

    A． 8               B． 3              C． 1            D． 7

40．有以下程序

#include "string.h"

#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

41．有以下程序

void ss( char  \*s, char  t)

{

while (\*s)

 {   if(\*s==t) \*s=t-'a'+'A';

         s++;

}

}

main()

{

char  str[100]="abcddfefdbd", c='d';

 ss(str, c);

 printf("%s\n", str1);

}

程序运行后的输出结果是（）。

A．ABCDDEFEDBD    B．abcDDfefDbD     C． abcAAfefAbA    D． Abcddfefdbd

42．以下程序调用findmax函数返回数组中的最大值。在下面划线处应填入的是（）。

#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);

}

main()

{

int x[5]={12,21,13,6,18};

  printf("%d\n",findmax(x,5));

}

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

43．有以下程序

#include "stdio.h"

#include "malloc.h"

main()

{

char \*q, \*p;

  p=(char\*) malloc(sizeof(char) \*20);    /\*为指针p分配一个地址\*/

  q=p;

  scanf("%s%s", p, q);

  printf("%s %s\n", p, q);

}

若从键盘输入：abc  def↙，则输出结果是：

A． def  def         B． abc  def         C． abc  d          D． d  d

44．下面程序的运行结果是（）。

   #include "stdio.h"

   #include "string.h"

   fun( char \*s)

{

char t[10];

s=t;

strcpy(t, "example");

}

   main()

{

char \*s;

fun(s);

puts(s);

    }

A．example└┘└┘└┘B．example└┘└┘C．example        D．不确定的值

45．下列程序段的输出结果是（）。

#include "stdio.h"

void fun( int \*x, int \*y)

 {

printf("%d%d",\*x, \*y);

\*x=3;

\*y=4;

}

main()

{

int x=1, y=2;

 fun(&y, &x);

 printf("%d %d", x, y);

}

A． 2 1 4 3           B． 1 2 1 2          C． 1 2 3 4           D． 2 1 1 2

46．下列程序的输出结果是（）。

#include "stdio.h"

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

47．有以下程序

#include "stdio.h"

#include "string.h"

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

{

int m, length=0;

for (m=1;m<argc; m++) length+=strlen(argv[m]);

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

}

程序编译连接后生成的可执行文件是file.exe，若执行时输入带参数的命令行是：

file  1234  567  89↙

则运行结果是（）。

A． 22              B． 17               C． 12             D． 9

48．有以下函数：

char \*fun(char \*s)

{   …

  return s;

}

该函数的返回值是（）。

A． 无确定值                            B．形参s中存放的地址值

C． 一个临时存储单元的地址              D． 形参s自身的地址值

49．假定下列程序的可执行文件名为file.exe，则在该程序所在的子目录下输入命令行：

          file  girl  boy↙

后，程序的输出结果是（）。

#include "stdio.h"

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

{

int m;

if (argc<=0) return;

  for (m=1; m<argc ; m++)

      printf("%c",\*argv[m]);

}

A． girl  boy       B． gb             C． gir              D． girlboy

50．设有一个名为file的C源程序，且已知命令行为：file  girl boy  student，则可得到以下运行结果的C源程序为（）。

girl

   boy

   student

   A． main( int argc, char \*argv[])

{

while (--argc>1)

printf("%s%c", \*argv, (argc>1)? '\n'：' ');

      }

   B． main( int a, char \* b[])

    {

 while(a-->1)

printf("%s\n", \*++b);

}

   C． main( int argc, char \*argv[])

     {

 while (++argc>0)

printf("%s%c",\*++argv, (argc>1) ? ' ' ：'\n');

}

   D． main(int argc, char \*argv[])

    {

 while (argc>1)

printf("%s",\*++argv);

}

7．2 填空题

1． 设有定义：int a, \*p=&a; 以下语句将利用指针变量p读写变量a中的内容，请将语句补充完整。

scanf("%d", 【p】 );

printf("%d\n", 【\*p】 );

2． 请填空：

建立如图所示存储结构所需的说明语句是【char \*p,c;】。

建立如图所示给c输入数据的输入语句是【scanf(“%c”,&c);】。

建立如图所示存储结构所需的赋值语句是【p=&c;】。

3．以下程序的运行结果是\_\_\_\_\_\_\_\_。

#include "stdio.h"

#include "string.h"

int \*p;

main()

{

int x=1, y=2, z=3;

  p=&y;

  fun(x+z, &y);

  printf("(1)%d  %d %d\n", x, y, \*p);  //2 6 6

}

fun( int x, int \*y)

 {

int z=4;

\*p=\*y+z;

  x=\*p-z;

  printf("(2)%d   %d  %d\n", x, \*y, \*p); //1 6 6

}

4． 下面程序段是把从终端读入的一行字符作为字符串放在字符数组中，然后输出。请填空。

    #include "stdio.h"

    #include "string.h"

    main()

    {

int m;

        char s[80], \*t;

        for (m=0; m<79; m++)

        {

 s[m]=getchar();

           if(s[m]=='\n')  break;

        }

        s[m]= 【’\0’】;

        t=【s】;

        while (\*t) putchar(\*t++);

    }

5． 下面程序段的运行结果是\_\_ eXAMPLE \_\_\_\_\_\_。

    char s[80],\*t="EXAMPLE";

    t=strcpy(s,t);

    s[0]='e';

    puts(t);

6． 函数sstrcmp()的功能是对两个字符串进行比较。当s所指字符串相等时，返回值为0；当s所指字符串大于t所指字符串时，返回值大于0；当s所指字符串小于t所指字符串时，返回值小于0（功能等同于库函数strcmp()）。请填空。

#include "stdio.h"

int  sstrcmp( char \*s, char \*t)

{

while (\*s && \*t && \*s==【\*t】)

   {

s++;

      t++;

}

return 【\*s-\*t】;

}

7． 下面程序的运行结果是\_\_35\_\_\_\_\_\_。

void swap(int \*a, int \*b)

{

 int\*t;

  t=a;

  a=b;

  b=t;

}

main()

{

int x=3, y=5, \*p=&x, \*q=&y;

 swap(p,q);

 printf("%d  %d\n",\*p, \*q);

}

8． 以下程序的输出结果是\_\_efgh\_\_\_\_\_\_。

#include "stdio.h"

main()

{

char \*p="abcdefgh",\*r;

  long \*q;

q=(long \*) p;

  q++;

r=(char \*) q;

 printf("%s\n", r);

}

9．下面程序的功能是将字符串中的数字字符删除后输出。请填空。

   #include "stdio.h"

   #include "malloc.h"

   void delnum( char \*t)

{

 int m, n;

      for (m=0,n=0; t[m]!='\0';m++)

if (t[m]<'0' || t[m]>'9')

{ t[n]=t[m]; n++;}

       【t[n]=0;】;

     }

     main()

{

char \*s;

      s=(char \*) malloc (sizeof(char));   /\*给s分配一个地址\*/

printf("\ninput the original string:");

gets(s);

delnum(s);

puts(【s】);

}

10．下面程序的功能是比较两个字符串是否相等，若相等则返回1，否则返回0。请填空。

#include "stdio.h"

   #include "string.h"

fun (char \*s, char \*t)

{

int m=0;

while (\*(s+m)==\*(t+m) && 【sum!=0】) m++;

return (【(\*(s+m)== '\0' && \*(t+m)=='\0')? 1：0】);

}

11．下面程序用来计算一个英文句子中最长单词的长度（字母个数）max。假设该英文句子中只含有字母和空格，在空格之间连续的字母串称为单词，句子以'.'为结束。请填空。

    #include "stdio.h"

    main()

    {

 static char s[]={" you make me happy when days are grey."}, \*t;

       int max=0, length=0;

       t=s;

       while (\*t!='.')

       {

 while (((\*t<='Z')&&(\*t>='A'))||((\*t<='z')&&(\*t>='a')))

           {

   length++;

       【t++】;

    }

  if (max<length) 【max=length】;

  length=0;

  t++;

}

printf("max=%d", max);

}

12．下面程序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"

main()

{

 char s[80], \*t1, \*t2;

   int m;

   gets(s);

   m=strlen(s);

   t1=s;

   t2=【s+m-1】;

  while(t1<t2)

   {  if (\*t1!=\*t2)  break;

      else { t1++;

【t2--】;}

}

       if (t1<t2) printf("NO\n");

       else printf("YES\n");

    }

13．当运行以下程序时，从键盘输入：apple↙

tample↙

    则下面程序的运行结果是\_\_  ample\_\_\_\_\_\_。

#include "stdio.h"

main()

{

char s[80], \*t;

  t=s;

 gets(t);

 while (\*(++t)!='\0')

       if (\*t=='a') break;

        else { t++;  gets(t); }

 puts(t);

}

14．当运行以下程序时，从键盘输入6↙，则下面程序的运行结果是\_\_\_\_\_976531\_\_\_。

#include "stdio.h"

#include "string.h"

main()

{

char s[]="97531", c;

  c=getchar();

  f(s,c);

  puts(s);

}

f(char \*t, char ch)

{

 while (\*(t++)!='\0');

   while（\*（t-1）<ch）

        \*(t--)=\*(t-1);

   \*(t--)=ch;

}

15．若有定义：int a[]={1,2,3,4,5,6,7,8,9,10,11,12}, \*p[3], m; 则下面程序段的输出是\_\_\_11\_\_\_\_\_。

   for ( m=0; m<3; m++) p[m]=&a[m\*4];

   printf("%d\n", p[2][2]);

16．下面程序的运行结果是\_\_\_\_2, 2\_\_\_\_。

#include "stdio.h"

main()

{

char s[]="1357", \*t;

  t=s;

printf("%c, %c\n", \*t, ++\*t);

}

17．以下程序将数组a中的数据按逆序存放。请填空。

#include "stdio.h"

#define M 10

   main()

{

int a[M], m, n, temp;

for( m=0; m<M; m++) scanf ("%d", a+m);

m=0;

n=M-1;

while(m<n)

 {

temp=\*(a+m);

   【\*(a+m)=\*(a+n)】;

   \*(【a+n】)=temp;

    m++;

    n--;}

 for (m=0;m<M;m++) printf("%3d",\*(a+m));

}

18．以下程序在a数组中查找与x值相同的元素的所在位置。请填空。

#include "stdio.h"

main()

{

int a[11], x, m;

printf("pleaseinput ten numbers:\n");

for(m=1;m<11;m++) scanf("%d", a+m);

printf("pleaseinput x:");

scanf("%d", &x);

\*a=【x】;

m=10;

while (x!=\*(a+m))

    【m--】;

if (m>0) printf("%5d's position is : %4d\n", x, m);

else printf("%d notbeen found!\n", x);

}

19．以下程序的功能是\_\_\_\_ PASCAL

         FORTRAN

          COBOL

          BASIC

\_\_\_\_。

#include "stdio.h"

main()

{

char \* s[]=={ "PASCAL", "FORTRAN", "COBOL", "BASIC"};

char \*\*p;

int n;

p=s;

for (n=0;n<4;n++)

  printf("%s\n", \*(p++));

}

20．若有定义和语句：int a[4]={1，2，3，4}，\*p; p=&a[2]; ，则\*--p的值是\_\_\_\_2\_\_\_\_。

21．若有定义和语句：int a[2][3]={0}, (\*p)[3];  p=a;，则p+1表示数组\_\_\_&a[1][0]\_\_\_\_\_。

22．若有如下定义和语句：

int \*p[3], a[6], n;

for(m=0;m<3;m++) p[m]=&a[2\*m];

则\*p[0]引用的是a数组元素【a[0]】；\*(p[1]+1)引用的是a数组元素【a[3]】。

23．若有以下定义和语句，在程序中引用数组元素a[m]的四种形式是：【\*(p+m】、【p[m]】、【\*(a+m)】和a[m]。（假设m 已正确说明并赋值）

    int a[10], \*p;

    p=a;

24．下面程序的输出结果是\_\_\_1   3    5    7   9    11\_\_\_\_\_。

   #include "stdio.h"

main()

{

int b[2][3]={1,3,5,7,9,11};

      int\*a[2][3];

      int i,j;

      int \*\*p, m;

     for(i=0;i<2;i++)

        for(j=0;j<3;j++)

            a[i][j]=\*(b+i)+j;

      p=a[0];

     for(m=0;m<6;m++)

        {

printf("%4d", \*\*p);

          p++;

        }

}

25．定义语句int \*f();和int (\*f)();的含义分别为【函数的返回值为指向int类型的指针】和【定义一个指向函数的指针】。

26．请根据运行结果，完成main函数中的填空。

    Array\_add( inta[], int n)

    {

int m, sum=0;

      for(m=0;m<n;m++)  sum+=a[m];

      return(sum);

    }

    main()

    {

intArray\_add(int a[], int n);

      static inta[3][4]={2,4,6,8,10,12,14,16,18,20,22,24};

      int \*p,total1, total2;

      【int (\*pt)()   】;

     pt=Array\_add;

      p=a[0];

     total1=Array\_add(p,12);

     total2=(\*pt)(【p,12】);

      printf("total1=%d\ntotal2=%d\n", total1,total2);

    }

    运行结果：total1=156

             total2=156