对下面问题 1 到问题 8 进行回答。
请注明标记表的对应括号中的题号答案数列。

问 1 根据C语言的规定对下面的表述，进行正确，错误判断。

（1）printf 函数输出的字符数目。

（2）浮点类型有 float，double，long double 三种精度，由 ANSI 标准指定的大小。

（3）指针变量不能被声明为 const。

（4）typedef 是由编译器解释，＃定义是由预处理解释。

（5）cast 是明示类型转换。指针变量也可以 cast

（6）当在一个局部变量和全局变量同名时，同在两个作用域范围内，全局变量是优先。

（7）在使用多个逻辑运算符的条件表达式中，真伪的判定是由右至左进行。
（8）对 float 型和 double 型，不可以使用位运算符。

下面是（1）到（8）的解答群：

ア 正确　　イ 错误

问 2 从解答群中选择适当的内容填在方框中。

枚举是列出数据枚举常量的所有可能值名称的数据类型，定义如下。
（9）枚举变量 {枚举常量列表} 枚举标记;
枚举常量的内部表现值为（10）型

<程序>

```
#include <stdio.h>
main()
{
    (9)  col {cyan, magenta, yellow = 6, black} color;
    int sum = 0, i;
    color = magenta;
    for (i = cyan; i <= black; i++)
        sum++;
        printf("%d", cyan);                    ... (11)
        printf("¥n%d", black);          ... (12)
        printf("¥n%d", color);          ... (13)
        printf("¥n%d", sum);            ... (14)
}
```

(9),(10)的解答群

ア char イ enum ウ extern エ int オ static

(11),(13)的解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

(12),(14)的解答群

ア 3 イ 4 ウ 6 エ 7 オ 8

问 3　阅读下面的说明，在程序中的方框中填入适当的内容

下面是将变量 dat_1 的低 8 位变为高 8 位，变量 dat_2 的高 8 位变为低 8 位 的二进制位反转后表示的程序。

《程序》

```
#include <stdio.h>
    main()
    {
```

```
unsigned short dat_1 = 0xABCD;
unsigned short dat_2 = 0x1234;

        dat_1 [OBJ]（15）8;
        dat_2 [OBJ]（16）8;
        dat_1 [OBJ]（17）[OBJ]dat_2;
        dat_2 =　　（18）[OBJ]dat_1;
        printf("%04hX", dat_2);　　.....（19）
[OBJ]}
```

(15)~(17)的解答群

ア <<=　　イ >>=　　ウ &=　　エ |=　　オ *=

(18)的解答群

ア &　　イ ^　　ウ ~　　エ !　　オ |

(19)的解答群

ア 12CD　　イ 32ED　　ウ 4402　　エ CD12　　オ ED32

问 4 关于结构体的表述，从下面的解答群里选择适当的内容填入方框中

下面是定义了一个日期的结构体。

```
[OBJ]（20）　hizuke　{
        int year;　　/* 年 */
        int month;　　/* 月 */
[OBJ]   int day;　　/* 日 */
} date_1, *date_2 = &date_1;
```

这是一个（21）的结构体。（22）被省略了。在这个结构体里变量 date_1 被设定时，关于年的表述正确的是（23），月的表述正确的是（24），日的表述正确的是（25）

(20)的解答群
ア #define　イ struct　ウ typedef　エ union
オ void

(21)的解答群
ア 结构名　イ 结构变量　ウ 结构定义　エ 指针变量　オ 成员名

(22)的解答群
ア 可能　イ 不可能

(23)的解答群
ア hizuke.yearイ date_1.year ウ date_1->year

エ date_2.yearオ *date_2.year

(24)的解答群
ア hizuke.month　イ date_1->month　ウ *date_1.month　エ *date_2.month

オ (*date_2).month

(25)的解答群
ア hizuke.day　イ date_1->day　ウ *date_1.day　エ date_2->day

オ *date_2.day

问 5　下面的程序执行时，请从解答群里选择（26）-（30）输出值

```
#include <stdio.h>
main() {
        char mark_1[][5] = {"zyx", "wv", "utsr", "qpo", "nmlk"};
        char *mark_2[5] = {"cba", "ed", "ihgf", "lkj", "ponm"};
        char *pt_1, **pt_2;

          pt_1 = *mark_1;
          printf("%s¥n",mark_1[1] + 1);              ...(26)
          printf("%s¥n",pt_1 + 1);                   ...(27)
          pt_2 = mark_2;
          printf("%s¥n",*pt_2 + 2);                  ...(28)
          printf("%c¥n",**pt_2 + 2);                 ...(29)
          printf("%s¥n",*(pt_2 + 2) + 2);            ...(30)
        }
```

(26)的解答群

ア zyx　　イ yx　　ウ wv　　エ v　　オ utsr

(27)的解答群

ア z　　イ yx　　ウ wv　　エ zx　　オ v

(28)的解答群

ア a　　イ cba　　ウ c　　エ ed　　オ ihgf

(29)的解答群

ア c　　イ d　　ウ e　　エ f　　オ ihgf

(30)的解答群

ア e　　イ gf　　ウ i　　エ ihgf　　オ ponm

问 6　请选择和下面函数同等的标准库函数

```
(31)int func_a(int ch)
        {
            if ( isalpha(ch) || isdigit(ch) )
                return 1;
            else
return 0;
}
```

解答群

ア isalnum イ isalpha ウ isdigit エ islower オ isupper

```
(32)int func_b(int ch) {
            ch = tolower(ch);
            if ( isdigit(ch) || ch >= 'a' && ch <= 'f' )
                return 1;
          else
return 0;
}
```

解答群

ア isalnum イ isalpha ウ isdigit エ ispunct オ isxdigit

```
(33)char *func_c(char *s1, const char *s2)
    {
          char *s3 = s1;
          for ( ; *s2; s2++ )
              *s1++ = *s2;
```

```
*s1 = '¥0';
return s3;
}
```

解答群

ア strcat イ strchr ウ strcmp エ strcpy オ strlen

```
(34)char *func_d(char *s1, const char *s2) {
    char *s3 = s1;
    for ( ; *s1 != '¥0'; s1++ );
    for ( ; *s2 != '¥0'; s1++, s2++ )
                *s1 = *s2;
    *s1 = '¥0';
     return s3;
}
```

解答群

ア strcat イ strchr ウ strcmp エ strcpy オ strlen

```
(35)char *func_e(char *s1, const char *s2, size_t n)
{
      char *s3 = s1;
      for ( ; n > 0; n-- )
            if ( *s2 == '¥0' )
                  *s1++ = '¥0';
            else
                  *s1++ = *s2++;
      return s3;
}
```

解答群

ア strncat イ strncmp ウ strncpy エ strstr オ sscanf

问 7 阅读下面有关程序的说明，在程序的方框中选入合适的内容

《程序说明》

这是一个学习四字成语的程序。

登陆的四字成语中随机有 1 个字被隐藏表示，这个被隐藏的字通过键盘输入，判断正误。共 10 个问题，随机选择四字成语，相同的成语不得成重复出现 。

《处理顺序》

① 从已注册的成语中，随机决定采用某个成语，如果该成语，已经被用过，则重新随机获取新的成语，直到未出现过的问题选择。
② 复制被选中的成语，以免原始数据被破坏。
③ 由随机抽出的隐藏的文字，把该文字复制到预先准备好的数组中。
④ 针对复制好的问题，隐藏随机显示的文字。（隐藏文字显示成'？'）
⑤ 从键盘输入的解答与答案比较，是否一致来判断正误。

⑥ 重复①～⑤的处理，直到 10 个问题全部表示。

<运行結果>
問 1:無?夢中 解答=我 結果...○
問 2:弱肉?食 解答=強 結果...○
問 3:絶?絶命 解答=体 結果...×
……
問 10:不言?行 解答=実 結果...○

<程序>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define TOI 12                    /*  登録されている四字熟語の数  */
main() {
char *word[TOI] = {"大同小異", "四面楚歌", "針小棒大", "五里霧中", "弱肉強食", "異口同音", "
絶対絶命", "起承転結", "大胆不敵","不言実行", "半信半疑", "無我夢中"};

char mon[9];              /*  储藏用于出題的四字词语  */
char kai[128];            /*  储藏入力答案  */
char sei[2];                      /*  储藏答案*/
char □(36)□;
int out[TOI];             /*  提出问题的状况登录  */
int r1, r2, i, q = 1;

for (i = 0; i < TOI; i++)
out[i] = 0;

while (q <= 10) {                                          /*  出题数设定为 10 问  */
          r1 = rand() % 4;
          do {
               r2 = rand() % TOI;
                 } while ( □ (37)  □);
                 out[r2] = 1;
              (38)      ;                                    /*  复制问题  */
          strncpy(sei, m_t + r1 * 2, 2);          /*  答案做成  */
                    (39)    ;                                          /*  空缺做成  */
                 printf("¥n 問%d:%s¥n", q, mon);
                 printf("解答=");
                 scanf("%s", kai);
```

```
                printf("結果...");
                if ( (40) )                                    /* 做出判定 */
                        printf("○¥n");
                else
                        printf("×¥n");
                q = q + 1;
        }
}
```

(36)的解答群
ア m_t=mon　イ m_t=*mon　ウ *m_t=mon　エ *m_t=*mon
(37)的解答群
ア out[r2] != 0　イ out[r2] != 1　ウ out[r2] == 0　エ out[r2] == 2
(38)的解答群
ア strcpy(mon, word[r2])　イ strcpy(word[r2], mon)
ウ strncpy(mon, word[r2], 8)　エ strncpy(word[r2], mon, 8)
(39)的解答群
ア strcpy(m_t + r1, "?")
イ strcpy(m_t + r1 * 2, "?")
ウ strncpy(m_t + r1 * 2, "?", 2)
エ strncpy((m_t + r1) * 2, "?", 2)
(40)的解答群
ア strcmp(kai, sei) != 0
イ strcmp(kai, sei) == 0
ウ strncmp(kai, sei, 2) != 0
エ strncmp(kai, sei, 2) == 0

问 8　阅读下面的程序在方框中填入合适的内容

《程序说明》
这个程序的目的是将文本文件（word.dat）中的符号【@】替换成 4 个空格表示并且，文本文件不满 30 行，每行文字 255 个以内。

文本文件（word.dat）
a@bc@def@ghij@k
lmn@op@qr
@st@u@vwxyz
ABC@DE@@FGI
HIJ@KLMNOP@QR
STUVWXYZ

变化结果显示为
a bc def　ghij　k
lmn op qr
　　st u　vwxyz
ABC DE　　FGI
HIJ KLMNOP QR
STUVWXYZ

记号变换
1，字符串的先头开始确认，将@记号替换成间隔文字，@以外的文字原样输出
2，@记号出现时，最近的 4 的倍数＋1 作为下一个字符的位置，把@记号替换成一个以上的间隔文字。

```
文字位置 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21
  展開前 a  @  b  c  @  d  e  f  @  g  h  i  j  @  k
                                    ↓
  展開後 a  △  △  △  b  c  △  △  d  e  f  △  g  h  i  j  △  △  △  △  k
```

【注】上述的△为半角的空白文字。

《运行顺序》

①读取文本文件，将文件保存在 2 维数组中

② 对保存的文件的@进行编辑

对 2 维数组,以行为单位进行搜索

从前往后检索，保存编辑后的数组

记号@出现时，填写空格直到最近的 4 的倍数的位置。记号@以外的文字原样记录到新的数组中将新的数组替换到 2 维数组中

③显示新的文件

《程序》

```
#include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
    int file_read(void);                /*  文件读入  */
    void henkan(int cnt);               /*  记号替换  */

    char data[30][255 * 4 + 1]; /*  保存用数组  */

main() {
        int cnt;                                /*  行数  */
        int i;
cnt = file_read();                              /*  文件读入  */
```

```
henkan(cnt);                    /*  记
号替换 */

/* 画面表示 */
for (i = 0; i < cnt; i++)
            printf("%s", data[i]);
}
/* 文本文件读入 */
int file_read(void)
{
        FILE *fp;
        int cnt = 0;
    if ((fp = fopen("word.dat", "r")) == NULL) {
        printf("读入的文件无法打开。¥n");
        exit(1);
    }
    while (fgets( OBJ (41) OBJ, 256, fp) != NULL) ;
        fclose(fp);
        return cnt;
}
/* 记号 @ 替换 */
void henkan(int cnt)
{
     char buf[255 * 4 + 1];
     int i, j, k, tb;

     for (i = 0; i < cnt; i++) {
            for (j = 0, k = 0; OBJ(42)OBJOBJ; j++) {
                if (data[i][j] == '@') {
                    tb = OBJ(43)OBJOBJ;
                    for ( ; k < tb; k++)
                    buf[k] = ' ';
                }
```

```
                    else
                        (44);
                }
                (45);
                strcpy(data[i], buf);
        }
}
```


(41)的解答群

ア data[--cnt] イ data[cnt] ウ data[++cnt] エ data[cnt++]

(42)的解答群

ア data[i][j] != '¥0'
イ data[i][j] == '¥0'
ウ data[j][i] != '¥0'
エ data[j][i] == '¥0'

(43)的解答群

ア k/4*4 イ (k/4+1)*4 ウ k/4*4+1 エ (k/4+1)*4+1

(44)的解答群

ア buf[k] = data[i][j]
イ buf[k++] = data[i][j]
ウ data[i][j] = buf[k]
エ data[i][j] = buf[k++]

(45)的解答群

ア buf[i] = '¥0' イ buf[j] = '¥0' ウ buf[k] = '¥0' エ buf[tb] = '¥0'