

一、选择题

- 下列叙述中正确的是 ()
 - 在 C 语言程序中, `main` 函数必须放在其他函数的最前面;
 - 每个后缀 (扩展名) 为 `.c` 的 C 语言程序都可以单独编译;
 - 在 C 语言程序中, 只有包含 `main` 函数的文件才可以单独编译;
 - 每个后缀 (扩展名) 为 `.c` 的 C 语言程序都必须包含一个 `main` 函数
- 以下选项中哪一个合法的常量 (其中 `□` 表示空格) ()
 - `3□5□7`
 - `0X7b`
 - `1.2e0.45`
 - `2.75E`
- 以下选项中哪一个可以作为用户定义的变量来使用 ()
 - `int`
 - `_08`
 - `xy.ch`
 - `x4#`
- 下列选项中哪一个实现了程序设计的模块化 ()
 - 三种程序设计的基本结构
 - 丰富的数据类型
 - 函数
 - 定义符号常量和外部变量
- 下列选项中关于常量以及符号常量的叙述中错误的是 ()
 - 常量在程序运行过程中的值不能发生变化
 - 符号常量是用预编译指令 `#define` 定义的标识符
 - 定义符号常量必须指定其数据类型
 - 用预编译指令 `#define` 定义的符号没有数据类型之分
- 若有定义: `double a = 32; int i = 0, k = 8;` 则下述不符合 C 语言规范的选项是 ()
 - `i = i++, a++;`
 - `i = (a+k) <= (i+k);`
 - `i = a%7;`
 - `i = !a;`
- `switch` 语句的一般形式为: `switch(表达式){...}`, 以下关于“表达式”值的叙述中正确的选项是 ()
 - 必须是逻辑值 `true` 或 `false`
 - 必须是字符型的值
 - 必须是整数
 - 可以是任意合法的数值
- `for` 语句实现循环的基本形式为: `for(表达式 1; 表达式 2; 表达式 3){ 循环体 }`, 下面关于表达式 2 的叙述中正确的是 ()
 - 必须是条件表达式
 - 必须是逻辑表达式
 - 表达式 2 是不可或缺的
 - 可以是任意表达式也可以没有
- 若 `a` 为数值类型, 则表达式 `(a == 1) || (a != 1)` 的值为 ()
 - 1
 - 0
 - 2
 - 不能确定
- 有定义语句 `char s[20] = "China"`, 则执行 `sizeof(s)` 和 `strlen(s)` 的结果是 ()。
 - 20, 20
 - 5, 5
 - 20, 5
 - 5, 20
- 下列表达式的值为“假”的选项是 ()
 - `(2 + 7) > 8`
 - `(3 == 4) > 2`
 - `2 > 3 || 7`
 - `3 > 2 && 7`
- 有定义语句 `int m;` 关于如下 `for` 语句的叙述中正确的是 ()

```
for(m = 8; m = 7; m++) printf("China\n");
```

 - 循环体执行 0 次
 - 循环体执行 1 次
 - 循环体执行 15 次
 - 构成无限循环

13. 有以下程序段

```
#include <stdio.h>
int main()
{ .....
    while(getchar() != '\0');
    .....
}
```

以下叙述正确的是 ()。

- A. 此 while 语句构成无限循环
- B. getchar() 不能出现在 while 语句的条件表达式中
- C. 当执行此 while 语句时, 只有按回车键程序才能终止循环的执行
- D. 当执行此 while 语句时, 只有按数字 0 键程序才能终止循环的执行

14. 若有定义语句 `int a[3][4], (*p)[3], *q;` 则以下正确的赋值语句是 ()

- A. `q = a;`
- B. `p = a;`
- C. `q = a[0];`
- D. `p = a[0];`

15. 设 `p` 是指针变量, 语句 `p = NULL;` 等价于 ()

- A. `p = "";`
- B. `p = "0";`
- C. `p = 0;`
- D. `p = '0';`

16. 以下选项中符合 C 语言规范的是 ()

- A. `char str[] = {"a", "b", "c", "d", "\0"};`
- B. `char *str = "I Love China!";`
- C. `char str[20]; str = "Hello World";`
- D. `char s1[10], s2[10] = "C"; s1 = s2;`

17. 若有如下语句: `int a, b; scanf("%d%d", &a, &b);` 以下选项中不能把 1 赋给变量 `a`, 2 赋给变量 `b` 的是 (其中□表示空格) ()

- A. 1, 2
- B. 1□2□3
- C. 1□2
- D. 1□2.3

18. 若有以下说明和定义

```
union data{
    int a;
    char c;
    double x;
} dt;
```

则下述叙述中错误的是 ()

- A. `data` 是一种数据类型
- B. 变量 `dt` 所占内存字节数与成员 `x` 所占字节数相同
- C. `union data` 是一种数据类型
- D. `&dt, &dt.a, &dt.c, &dt.x` 具有相同的值

19. 有如下条件表达式 `(!flag) ? 'a' - 32 : 'a';` 下列选项中等价于表达式 `(!flag)` 的是 ()

- A. `(flag == 1)`
- B. `(flag != 1)`
- C. `(flag == 0)`
- D. `(flag != 0)`

20. 有以下程序

```
#include <stdio.h>
int main()
{ int x = 7;
  printf("%d\n", ++x);
  return 0;
}
```

程序运行后的输出结果为 ()

- A. 7
- B. 8
- C. 9
- D. 10

21. 有以下程序

```
#include <stdio.h>
int func(char *s){
    int count = 0, i;
    for(i=0; *(s+i) != '\0'; i++){
        if(*(s+i) >= '0' && *(s+i) <= '9') count++;
    }
    return count;
}
int main(){
    char *str = "1996 is a leap year, and February has 29 days."
    int n = func(str);
    printf("%d\n", n);
    return 0;
}
```

则程序运行后的输出结果是 ()

A. 10 B. 46 C. 6 D. 9

22. 有以下程序 (其中□表示空格):

```
#include <stdio.h>
void func(int *p, int n){
    int i, j, k;
    for(i=1; i<n; i++){
        int temp = *(p+i);
        j = i-1;
        while(j>=0){
            if(temp >= *(p+j)) break;
            *(p+j+1) = *(p+j);
            j--;
        }
        *(p+j+1) = temp;
    }
}

int main(){
    int a[10] = {1,2,3,4,5,6,7,8,9,0}, i;
    func(a, 10);
    for(i=0; i<10; i++) printf("%d□", a[i]);
    printf("\n");
    return 0;
}
```

上述程序的运行结果为 ()

A. 1□2□3□4□5□6□7□8□9□0 B. 9□8□7□6□5□4□3□2□1□0
C. 0□1□2□3□4□5□6□7□8□9 D. 0□9□8□7□6□5□4□3□2□1

23. 有以下程序:

```
#include <stdio.h>
char *func(char *s, char c1, char c2){
    int i;
    for(i=0; *(s+i) != '\0'; i++){
        if(*(s+i) == c1) *(s+i) = c2;
    }
    return s;
}
int main()
{
    char s[] = "food";
    func(s, 'o', 'e');
}
```

```

    puts(s);
    return 0;
}

```

程序运行后的输出结果是 ()。

- A. food B. foot C. feed D. foed

24. 有以下程序段 (其中□表示空格):

```

#include <stdio.h>
int func(int a, int b){
    if(a > b){
        int t = a; a = b; b = t;
    }
    int r = b%a;
    while( r ){
        b = a;      a = r;      r = b%a;
    }
    return a;
}
int main()
{
    int m, n, c;
    scanf("%d%d", &m, &n);
    c = func(m, n);
    printf("%d\n", c);
    return 0;
}

```

若程序运行时输入: 8□6<回车>, 则程序运行结果为 ()

- A. 8 B. 6 C. 24 D. 2

25. 有以下程序

```

void func(char *s, char *ss, int n1, int n2)
{
    int i, j;
    for(i=n1, j=0; i<n1+n2 && s[i] != 0; i++, j++) ss[j] = s[i];
    ss[j] = 0;
}
int main()
{
    char s1[20]="I Love China", s2[20];
    func(s1, s2, 2, 4);
    printf("%s\n", s2);
    return 0;
}

```

以上程序执行后, 运行结果是 ()

- A. Love China B. China C. Love D. hina

26. 有以下程序

```

#include <stdio.h>
int length(char *s){
    int i;
    for(i = 0; *(s+i) != 0; i++);
    return i;
}
int func(char *s){
    int len = length(s);
    int base = 1, count = 0;
    for(int i = len-1; i >= 0; i--){

```

```

        count += (*(s+i) - '0')*base;
        base = base * 10;
    }
    return count;
}
int main()
{
    char s[10];
    int num;
    gets(s);
    num = func(s);
    printf("%d\n", num);
    return 0;
}

```

程序运行后输入：14570<回车>，则程序运行输出结果的是（）

- A. 17 B. 5 C. 14570 D. 7541

27. 有以下程序，当程序运行时的输出是（其中□表示空格）（）

```

#include <stdio.h>
#include <math.h>
int func(int x){
    int i, flag = 0, m = sqrt(x);
    for(i = 2; i<=m; i++){
        if(x%i == 0)break;
    }
    if(i > m) flag = 1;
    return flag;
}
int main(){
    int k;
    for(k = 11; k < 20; k++){
        if(func(k)) printf("%d□", k);
    }
    printf("\n");
    return 0;
}

```

- A. 11□12□13□14□15□16□17□18□19□20 B. 11□13□15□17□19
C. 11□13□17□19 D. 12□14□16□18

28. 有以下程序

```

#include <stdio.h>
int func(int n)
{
    if(n == 1) return 1;
    return 2+func(n-1);
}
int main()
{
    int x = 5;
    x = func(x);
    printf("%d\n", x);
    return 0;
}

```

程序的运行结果是（）

- A. 9 B. 19 C. 25 D. 无输出

29. 有以下程序（其中□表示空格）

```
#include <stdio.h>
int func(char *s)
{
    int i, word = 0, num = 0;
    char c;
    for(i=0; (c = *(s+i)) != '\0'; i++){
        if(c == ' ') word = 0;
        else if(word == 0){
            word = 1;
            num++;
        }
    }
    return num;
}
int main()
{
    char s[] = "A□main□may□be□□□destroyed□□but□not□defeated";
    int num = func(s);
    printf("%d\n", num);
    return 0;
}
```

该程序的运行结果是（ ）

- A. 43 B. 10 C. 33 D. 8

30. 以下程序的运行结果是（ ）

```
#include <stdio.h>
void func(int *n)
{
    static int m=2;
    *n *= m-1;
    m += 2;
}
int main()
{
    int x=1, i;
    for(i=0; i<3; i++)
    {
        func(&x);
    }
    printf("%d\n", x);
    return 0;
}
```

- A) 8 B) 10 C) 11 D) 15

二、程序填空题

1. 以下是判断闰年的函数，请将程序补充完整

```
int leap(int year)
{
    int flag = 0;
    if (year%400 == 0 || (【1】))
        【2】 //不允许在此处使用 return 语句
    return flag;
}
```

2 以下函数是求两个正整数的最小公倍数的函数及其调用，请将程序补充完整；

```

int lcm(int x, int y)
{
    int i, m = x>y?x:y;
    for(i = m; 【3】 ; i++)
    {
        if(i%x == 0 && i%y == 0)
            【4】
    }
    return i;
}
int main()
{
    int m, n;
    scanf("%d%d", &m, &n);
    int c = 【5】 ;
    printf("%d 和 %d 的最小公倍数是 %d\n", m, n, c);
    return 0;
}

```

3. 向带头结点的链表第 i 个结点前插入一个新结点的函数如下，请将程序补充完整

```

typedef struct node{
    int data;
    struct node *next;
} LNode;
//h 表示链表头结点的指针，num 新结点数据域的值
void insertBefore(LNode *h, int i, int num)
{
    int pos = 1;
    LNode *p = h->next;
    LNode *node = (LNode*)malloc(sizeof(LNode));
    node->data = num;
    node->next = NULL;
    while (pos < i - 1 && p){
        pos++;
        【6】
    }
    if (p->next == NULL){
        p->next = node;
    }
    else{
        LNode *q = p->next;
        【7】
        【8】
    }
    return h;
}

```

4. 下面函数的功能是：将一个字符串中包含的所有整数（不考虑负数的情况）取出，并返回整数的个数，请将程序补充完整。其中 `stoi` 函数实现将一个仅包含数字的字符串转换为十进制整数，其函数原型为：

```

int stoi(char *s);
int numbers(char *s, int *p) //p 为存储所有整数的数组
{
    int i, j, k, count=0;
    for(i=0; s[i] != 0; i++)
    {
        if( 【9】 ){

```

```

        char numchars[10];
        k = 0;
        for(j = i; s[j] >= '0' && s[j] <= '9'; j++)
        {
            【10】
            k++;
        }
        numchars[k] = 0;
        p[count++] = stoi(numchars);
        i = j;
    }
}
return count;
}

```

三、读程序写结果

```

1. void Ex3_1(char str[])
{
    char *p=str,ch;
    for (int i = 0; i < strlen(str) / 2; i++)
    {
        ch = *(p + i);
        *(p + i) = *(p + strlen(str) - 1 - i);
        *(p + strlen(str) - 1 - i) = ch;
    }
}

```

```

int main()
{
    char str[20];
    printf("In the string:");
    Ex3_1(str);
    puts(str);
    return 0;
}

```

从键盘输入字符串：1234abba1234

```

2. int Ex3_2(char str[], char ch)
{
    int i,k;
    for (i = k = 0; str[i] != '\0'; i++)
        if (str[i] == ch || str[i] == ch + 32 || str[i] == ch - 32)
            k++;
    return k;
}

```

```

int main()
{
    char str[20],ch;
    gets(str);
    ch=getchar();
    printf("the \' %c\':%d",ch,Ex3_2(str,ch));
    return 0;
}

```

键盘输入：abcdABCDCba<回车>CD<回车>

```

3. void Ex3_3(char *s1, char *s2,int k)
{
    s1 = s1 + k - 1;
}

```



```

        while(*s2++ = *s1++);
    }

int main()
{
    char str1[20]="I Love The C Program",str2[20];
    Ex3_3(str1, str2, 3);
    puts(str1);
    puts(str2);
    return 0;
}

4. int *Ex3_4(int(*p)[4])
{
    int *pt=NULL;
    for (int i = 0; i < 4;i++)
        if ((*p + i) >= 90) pt = *p;
    return pt;
}

int main()
{
    int *s=NULL;
    int a[][4] = { { 76, 83, 87, 65 }, { 88, 92, 77, 53 }, { 98, 72, 83, 60 } };
    for (int j = 0; j < 3; j++)
    {
        s = Ex3_4(a + j);
        if (s != NULL)
            printf("%4d%4d%4d%4d\n", *s, *s++, *s++, *s++);
    }
    return 0;
}

```

四. 编程题

1. 设计一个函数实现如下功能：将一个正整数转换为十六进制表示的字符串，在 main 函数中输入一个正整数，然后调用该函数，输出函数返回结果。如输入：155，输出 9B。参考函数原型如下：char *convert(int x, char *s);

2. 编写一个函数实现如下功能：统计一个字符串中出现不同字符的个数，如字符串“ababc3x”中含有 5 个不同字符。建议函数原型如下：

```
int count(char *s);
```