

单选

下述正确的C语言常量是: (C)。

- A、E2
- B、5.0E
- C、3e-3
- D、1.5E2.5

若有定义: `int a=7; float x=2.5, y=4.7;`  
则表达式 `x+a%3*(int)(x+y)%2/4` 的值是 (D)。

- A、0.000000
- B、2.750000
- C、3.500000
- D、2.500000

根据下面的程序, 使 `i=123`, `c1='A'`, `c2='B'` 哪一种是正确的键盘输入方法 (C)。  
(规定用[CR]表示回车, U表示空格)

```
main()
{
    int i;
    char c1, c2;
    scanf("%d%c%c", &i, &c1, &c2);
}
```

- A、123AUB[CR]
- B、123UAUB[CR]
- C、123AB[CR]
- D、123[CR]AB[CR]

下列程序段的输出结果为 (B)。

```
int a=7, b=9, t; t=a*=a>b?a:b; printf("%d", t);
```

- A、49
- B、63
- C、7
- D、9

`int a=1, b=2, c=3; if(a>c)b=a;a=c;c=b;` 则 `c` 的值为 (C)。

- A、1
- B、3
- C、2
- D、不一定

`t` 为 `int` 类型, 进入下面的循环之前, `t` 的值为 0

```
while( t=1 )
{ .....}
```

则以下叙述中正确的是(C)。

- A、循环控制表达式的值为0
- B、循环控制表达式不合法
- C、循环控制表达式的值为1
- D、以上说法都不对

针对for循环语句, for(表达式1;表达式2;表达式3), 下列说法正确的是(B)。

- A、break语句可以一次性跳出多层嵌套循环
- B、break语句只能跳出最内层循环
- C、continue语句可以一次性跳出多层嵌套循环
- D、continue 语句可以跳出最内层循环

以下不正确的定义语句是 (C) 。

- A、double x[5]={2.0, 4.0, 6.0, 8.0, 10.0};
- B、char c1[]={'1','2','3','4','5'};
- C、int y[5]={0, 1, 3, 5, 7, 9};
- D、char c2[]={'\x10','\xa','\x8'};

以下正确的说法是(A)。

- A、函数的类型决定返回值的类型
- B、定义函数时,形参的类型说明可以放在函数体内
- C、return后面不能为表达式
- D、如果形参与实参的类型不一致,以实参类型为准

C语言规定, 函数返回值的类型是由 (A) 。

- A、在定义该函数时所指定的函数类型所决定
- B、return语句中的表达式类型所决定
- C、调用该函数时系统临时决定
- D、调用该函数时的主调函数类型所决定

函数fun用于将一字符串按相反次序显示, 则横线上的表达式为(D)。

```
void fun(char *s)
{ if(! *s)
    return;
    _____;
    printf("%c", *s);
}
```

- A、printf("%s", s+1);
- B、fun(\*(s+1));
- C、printf("%s", s+strlen(s)-1);
- D、fun(s+1);

在宏定义#define PI 3.14159中, 用宏名替换一个(A).

- A、字符序列
- B、函数名
- C、单精度数
- D、双精度数

若有说明:int \*p1,\*p2,m=5,n; 以下均是正确赋值语句的选项是 (A)。

- A、p1=&m;p2=p1;
- B、p1=&m;p2=&p1
- C、p1=&m;\*p2=\*p1;
- D、p1=&m;p2=&n;\*p1=\*p2;

下列程序的运行结果是: (B).

```
void fun(int *a, int *b)
{ int *k;
  k=a;a=b;b=k;
}
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、0      0
- D、编译出错

C语言结构体类型变量在程序执行期间 (B)。

- A、没有成员驻留在内存中
- B、所有成员一直驻留在内存中
- C、只有一个成员驻留在内存中
- D、部分成员驻留在内存中

判断

表达式 (j=3, j++) 的值是 4. F

若 a=3, b=2, c=1 则关系表达式“(a>b)==c” 的值为“真”。 T

C 语言的输入功能是由系统提供的输入函数实现的. T

char c[6]="abcde"; printf("%3s", c)表示输出的字段的宽度为 3 位, 如果被输出的数据的位数大于 3, 只输出 3 位数. F

条件表达式 `x?'a':'b'` 中, 若 `x=0` 时, 表达式的值为 `'a'`. F

设 `x=1, y=2, z=3`, 则逻辑表达式 `x-y>z&& y!=z` 的值为 0. T

已知 `a=3, b=4, c=5`. 则逻辑表达式 `a+b>c && b==c` 值为 0. T

若有说明 `int c`; 则 `while(c=getchar());` 没有语法错误. T

do-while 循环由 do 开始, while 结束, 循环体可能一次也不做. F

for 循环语句只能用于循环次数确定的情况下. F

若有定义 `int a[10]`; 则 `for (i=0; i<=10; i++) a[i]=i`; 编译时无语法错误. T

定义 `int x[5], n`; 则 `x=x+n`; 或 `x++`; 都是正确的. F

二维数组在内存中存贮时, 是按列的顺序连续存贮各元素的值. F

变量根据其作用域的范围可以分作局部变量和全局变量. T

通过 `return` 语句, 函数可以带回一个或一个以上的返回值. F

函数的返回值可以通过函数中的 `return` 语句获得. T

若有宏定义: `#define S(a,b) t=a;a=b;b=t` 由于变量 `t` 没定义, 所以此宏定义是错误的. F

定义 `int a[] = {1, 2, 3, 4}`, `y, *p=&a[1]`; 则执行 `y=(*--p)++` 后, `y` 的值是 2. F

一个共用体变量中可以同时存放其所有成员. F

用 `fopen("file", "r+")`; 打开的文件 `"file"` 可以进行修改. T

填空

若有定义: `char c='\010'`; 则变量 `C` 中包含的字符个数为\_\_\_\_. 1

`getchar()` 函数只能接收一个\_\_\_\_. 字符

执行下列语句后, `b` 的十进制值是\_\_\_\_. 1

```
int x=240, y=15, b;  
char z='A';  
b=(( x && y ) && ( z < 'a' ));
```

设x和y均为int型变量, 则以下for循环中的scanf语句最多可执行的次数是\_\_\_\_. 3

```
for (x=0, y=0; y!=123&& x<3; x++)
    scanf ("%d", &y);
```

C 语言中, 数组名是一个不可变的\_\_\_\_常量, 不能对它进行加减和赋值运算. 地址

函数调用语句 func((e1, e2), (e3, e4, e5)) 中含有\_\_\_\_个实参. 2

定义 int \*p, a[5]; 将数组 a 的首地址赋给指针变量 p 的语句是\_\_\_\_. p=a; p=&a[0];

结构体是不同数据类型的数据集合, 作为数据类型, 必须先说明结构体\_\_\_\_, 再说明结构体变量. 类型

```
/*-----
```

**【程序设计】**

```
-----
```

功能: 编写函数 sum(int m,int n), 求出 m~n 之间(包括 m 和 n,m<n)能同时被 3 和 5 整除的数的和, 并将这些数的和作为返回值。

```
-----*/
```

```
#include "stdio.h"
```

```
int sum(int m,int n)
```

```
{
```

```
    /******Begin******/
```

```
    int i, s=0;
```

```
    for(i=m;i<=n;i++)
```

```
        if(i%3==0 && i%5==0)
```

```
            s+=i;
```

```
    return s;
```

```
    /****** End ******/
```

```
}
```

```
main()
```

```
{
```

```
    void TestFunc();
```

```
    int m=15,n=300;
```

```
    printf("sum=%d\n",sum(m,n));
```

```
    TestFunc();
```

```
}
```

```

void TestFunc()
{
    FILE *IN,*OUT;
    int i,m,n;
    IN=fopen("in.dat","r");
    if (IN==NULL)
    {
        printf("Open File Error");
        return;
    }
    OUT=fopen("out.dat","w");
    if(OUT==NULL)
    {
        printf("Write File Error");
    }
    for(i=0;i<5;i++)
    { fscanf(IN,"%d%d",&m,&n);
      fprintf(OUT,"%d\n",sum(m,n));
    }
    fclose(IN);
    fclose(OUT);
}

```

/\*-----

### 【程序设计】

功能：求 5 行 5 列矩阵的主、副对角线上元素之和。注意，  
 两条对角线相交的元素只加一次。  
 例如：主函数中给出的矩阵的两条对角线的和为 45。

-----\*/

```

#include "stdio.h"

```

```

#define M 5

```

```

int fun(int a[M][M])

```

```

{

```

```

    /******Begin******/

```

```

    int i, s=0;

```

```

    for(i=0;i<M;i++)

```

```

        s+=a[i][i]+a[i][M-i-1];
    s-=a[(M-1)/2][(M-1)/2];
    return s;

    /*****      End      *****/

}

void TestFunc()
{
    FILE *IN,*OUT;
    int iIN[M][M],iOUT;
    int i,j;
    IN=fopen("in.dat","r");
    if(IN==NULL)
    {
        printf("Please Verify The Currenrt Dir..It May Be Changed");
    }
    OUT=fopen("out.dat","w");
    if(OUT==NULL)
    {
        printf("Please Verify The Current Dir.. It May Be Changed");
    }
    {   for(i=0;i<M;i++)
        for(j=0;j<M;j++)
            fscanf(IN,"%d",&iIN[i][j]);
        iOUT=fun(iIN);
        fprintf(OUT,"%d\n",iOUT);
    }
    fclose(IN);
    fclose(OUT);
}

main()
{
    int a[M][M]={ { 1,3,5,7,9},{2,4,6,8,10},{2,3,4,5,6},{4,5,6,7,8},{1,3,4,5,6} };
    int y;
    y=fun(a);
    printf("s=%d\n",y);
    TestFunc();
}

```

/\*-----

【程序填空】

-----

题目：求矩阵 a 的副对角线上元素之和。

-----\*/

```
#include "stdio.h"
main()
{
    int a[3][3]={ 1,3,5,7,9,2,4,6,8},i,j,sum=0;
    for(i=0;i<3;i++)
        /******FILL******/
        for(j=0;_____:j++)    //j<3
            /******FILL******/
            if(_____) sum=sum+a[i][j];    //i+j==2
            printf("%3d",sum);
}
```

/\*-----

**【程序填空】**

-----

题目：下面的程序利用字符串处理函数实现对字符串的处理。

-----\*/

```
#include "stdio.h"
/******FILL******/
____    //include <string.h>
main()
{
    char a[50]="abc",b[50]="ABC",*p1,*p2,str[50]="xyz";
    p1=a; p2=b;
    strcpy(str+2,strcat(p1,p2));
    /******FILL******/
    printf(____,str);    // "%s"
}
```

/\*-----

**【程序改错】**

-----

功能：输出 Fabonacci 数列的前 20 项,要求变量类型定义成浮点型，



输出时只输出整数部分，输出项数不得多于或少于 20。

```
-----*/
#include "stdio.h"
void fun()
{
    int i;
    float f1=1,f2=1,f3;
    /*****ERROR*****/
    printf("%8d",f1);    //printf("%8.0f%8.0f",f1, f2);
    /*****ERROR*****/
    for(i=1;i<=20;i++)    //for(i=3;i<=20;i++)
    {
        f3=f1+f2;
        /*****ERROR*****/
        f2=f1;    //f1=f2;
        /*****ERROR*****/
        f3=f2;    //f2=f3;
        printf("%8.0f",f1);    //printf("%8.0f",f3);
    }
    printf("\n");
}

main()
{
    fun();
}

/*-----
【程序改错】
-----
```

功能：求两数平方根之和，作为函数值返回。  
例如：输入 12 和 20，输出结果是：y = 7.936238。

```
-----*/
#include "stdio.h"
#include "conio.h"
#include "math.h"

/*****ERROR*****/
double fun (double *a, *b)    //double fun (double *a, double *b)
{
    double c;
```

```
/******ERROR*****/
c = sqrt(a)+sqrt(b) ; //c = sqrt(*a)+sqrt(*b);
/******ERROR*****/
return a; //return c;
}

main ( )
{
    double a, b, y;
    printf ( "Enter a & b : "); scanf ("%lf%lf", &a, &b );
    y = fun (&a, &b); printf ("y = %f \n", y );
}
```