

C_C++ § 言笔试面试题目含答案

C 语言

1. 语言

1.

```
void Example(char acHello[])
(
printf("%d", sizeof(acHello));
return;
}

void main()
(
char acHello[] = "hello";
Example(acHello);
return;
}
```

的输出是()

A 4 B 5 C 6 D 不确定

2. 语言的字序默认是大字序的 如果主机是小字序 在语言通信语言
行字序语言 如果主机是网网

大字序 保证了程序的一致性及可移植性 最好也在程序中加入字序语言的
操作; 空操作,。(

)

A. 正确 B. 错误

3. `int *(*ptr)();`

以下叙述中正确的是

A) ptr是指向一维数组的指针

B) ptr是指向 int 型数据的指针

C) ptr是指向函数的指针，该函数返回一个 int 型数据

D) ptr是指向函数的指针，该函数的返回值是指向 int 型数据的指针

4. 此函数把 32 位 IP 地址 (网络序) 以字符串的方式打印出来

```
*IpAddr2Str(unsigned long ulIpAddr)
```

```
{
```

```
char szIpAddr[32];
```

```
unsigned long ulLocIpAddr = ntohl(ulIpAddr); // 把网络序主机序
```

```
(void)VOS_sprintf(szIpAddr, "%d.%d.%d.%d", ulLocIpAddr >> 24,
```

```
(ulLocIpAddr >> 16) & 0xff, (ulLocIpAddr >> 8) & 0xff, ulLocIpAddr &
```

```
0xff);
```

```
return szIpAddr;
```

```
}
```

下面描述正确的是

A、 函数 szIpAddr 空不

B、 函数返回局部变量 szIpAddr 的地址

C、 语出的 IP 地址次序语倒

5.

```
#define OK 0#define ERR 1#define ERROR (-1)#define BUFFER_SIZE  
256char *GetMemory(unsigned long ulSize)
```

```

char *pcBuf = NULL;

pcBuf = (char *)malloc(ulSize);

if(NULL == pcBuf)
{
return ERROR;
}

return pcBuf;
}

void Test(void)
{
char *pszBuf = NULL;

pszBuf = GetMemory(BUFFER_SIZE);

if(NULL != pszBuf)
{
strcpy(pszBuf, "Hello World!\r\n");
printf(pszBuf);
free(pszBuf);
}

return;
}

```

如下描述正的是，确

A、 pszBuf 指向的存不能超语内 255

B、 GetMemory 函的常分支返回了数异-1 是一非法地址个 G GetMemory^ 常

分支有语放空语 异没

Dk pcBuf 语局部指语 指向的存在内将 GetMemory^ 语用后语放

```

6、 #include "stdio.h" unsigned short *Sum(unsigned char a, unsigned char b)
{
unsigned short s = 0;
s = a + b;
return &s;
}

int main()
{
unsigned short *p = NULL;
unsigned char a = 1, b = 2;
p = Sum(a, b);
printf("%u+%u", a, b);
printf("=%u\n", *p);
return 0;
}

```

程序运行结果是;

A. 1+2=0 B. 1+2=3 C. 1+2=NULL D. 不可言语 7、 语有如下定语:

```

BOOL gStatusA = FALSE; BOOL gStatusB = TRUE; int gVarA = 100;

```

```

int gVarB = 100;

```

语行 main 函数后数 gVarA 和 gVarB 的语分语语 ()

```

BOOL CheckA()

```

```

{

```

```

if(gStatusA)

```

```

(

```

```
gVarA++;  
}  
else  
(  
gVarA--;  
}  
return gStatusA;}  
  
BOOL CheckB()  
(  
if(gStatusB)  
(  
gVarB++;  
}  
else  
(  
gVarB--;  
}  
return gStatusB;}  
  
int main(int argc, char* argv[])  
(  
if(CheckA() && CheckB() )  
(  
printf( "Status OK " );  
}  
return 0;  
}
```

A, 99 和 100 B, 99和 101 C 99 和 99 D 101 和 998. 下面的代语中 函数 Test 语行
完语后 希望语出无符语整型的号 1。

```
void VarInit(unsigned char *pucArg)
{
    *pucArg = 1;
    return;
}

void Test()
{
    unsigned long ulGlobal;
    VarInit(&ulGlobal);
    printf("%lu", ulGlobal);
    return;
}
```

下面描述正的是, ;, 确

A. 语 VarInit() 函语语的语型语语数参数

B. printf;,语出格式语语

C. 语语 VarInit (中参数 pucArg 的语语空指语 9、

```
void AddFunc (unsigned int a, unsigned int b, unsigned int * c)
{

    *c = a + b
}
```

```

void main(void)
(
unsigned char e = 200;
unsigned char f = 100;
unsigned char g = 0;
AddFunc((unsigned int)e, (unsigned int)f, (unsigned int *)&g);
printf("%d ", g);
}

```

下面语法正的是;,, 确

A 语 g 语行语型语语语致函语用语存越界 数写内

B 语 e、f 语行语型语语语致函语用语存越界 数写内

C 函语用语不能改语数 g 的语。

10、 void main(void)

```

(
unsigned char a =200;
unsigned char b =100;
unsigned char c = 0;
c=a+b;
printf("%d %d", a+b, c);}

```

下列程序的语行语果语 ()

A 300 300

B 44 44

C 300 44

D 44 300

11、在 X86 VC++6.0 语境下 有下列程序#include <stdio.h>


```

int main()
{
char cA;

unsigned char ucB;

unsigned short usC;

cA = 128;

ucB = 128;

usC = cA + ucB;

printf("0x%x ", usC);

usC = cA + (short)ucB;

printf("0x%x ", usC);

usC = (unsigned char)cA + ucB;

printf("0x%x ", usC);

usC = cA + (char)ucB;

printf("0x%x ", usC);

return 0;
}

```

语出语果是()

A) 0x0 0x0 0x100 0xff00

B) 0x0 0x100 0x100 0xff00

C) 0x0 0x100 0x100 0x0

D) 0x0 0x0 0x100 0x012. switch(c)中的 c 的据语型可以是数 char、long、float unsigned、bool. ()

A. 正确 B. 语语

13. void Example()

{

```
int i;

char acNew[20];

for(i = 0; i < 10; i++)

{

acNew[i] = '0';

}

printf("%d\n", strlen(acNew));

return;

}
```

的语出语()

A 0 B 10 C 11 D不定确

14. 如下程序用于把"blue"字符串返回,

```
char *GetBlue(void)

{

char* pcColor ;

char* pcNewColor;

pcColor = "blue";

pcNewColor = (char*)malloc(strlen(pcColor));

if(NULL == pcNewColor)

(

return NULL;

}

strcpy(pcNewColor, pcColor);

return pcNewColor;

}
```

下面描述正的是，确

- A、 字符串“blue”存放在语 内
- B、 函数 GetBlue 返回局部变量地址
- C、 内存分配度不 内 strcpy函数考语越界 数

15. 语出以下定语:

```
char acX[ ]= "abcdefg";  
char acY[ ]= {'a','b','c','d','e','f','g'};
```

语正的述语确叙()

- A) 语数 acX 和语数 acY 等价 B) 语数 acX 和语数 acY 的语度相同
- C) 语数 acX 的语度大于语数 acY 的语度 D) 语数 acX 的语度小于语数 Y 的语 度 16.

有以下程序段

```
char acArr[]= "ABCDE";  
char *pcPtr;  
for(pcPtr = acArr; pcPtr < acArr + 5; pcPtr++)  
{  
printf("%s\n", pcPtr);  
}  
return;
```

语出语果是()

- A) ABCD B) A C) E D) ABCDE
- B D BCDE
- C C CDE

D B DE

E A E

```
17. void Example()
(
int i;
char acNew[20] = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0}

for(i = 0; i < 10; i++)
(
acNew[i] = '0';
}

printf("%d\n", strlen(acNew));

return;
}
```

的语出语, ()

A 0 B 10 C 11 D不定确

18. 如下程序用于把"blue"字符申打印出, 来

```
void PrintBlue()
(
char pcBlue[] = ('b', 'l', 'u', 'e');
printf("%s", pcBlue);
return;
}
```

下面描述正的是;, , 确

A、 pcBlue 语初始化语有字符申语束符 数没

B、 语数 pcBlue 定语语必语指明语度

19. 如下代语语语中 FuncA 语每毫秒定语语行的函 在语函中 需要语语每数
数 TIME_INTERVA 厘秒语行一次 DO_Something() 的操作。

语指出段代语中的语语,

```
#define ULONG unsigned long
```

```
#define TIME_INTERVAL 200
```

```
void DoSomething(void)
```

```
{
```

```
/*...*/
```

```
return;
```

```
}
```

```
void FuncA ( )
```

```
{
```

```
static ULONG ulPreCall = 0 ;
```

```
ULONG ulNowInMsHigh = 0 ;
```

```
ULONG ulNowInMsLow = 0 ;
```

(VOID) VOS_Tm_Now(&ulNowInMsHigh, &ulNowInMsLow) ; /* 语取前的语 语
当, 以毫秒语语位 用 64bits表示 ulNowInMsHigh 语高 32 位, ulNowInMsLow 语低
32 位*/

```
if( ( 0 == ulPreCall ) || ( ulNowInMsLow >= (ulPreCall +  
TIME_INTERVAL)))
```

```
{
```

```
ulPreCall = ulNowInMsLow;
```

```

}
else
{
return ;
}
DoSomething();
return ;
}

```

A、函数 FUNC_ 第一次被调用时不执行 Do_Something() 操作 B、函数

FUNCR 能在一段代码后失效 因为 ulNowInMsLow 溢出 翻

C、ulPreCall 环境语法定语语量 静

20、

```

#define NULL 0

#define MEM_OK 0

#define MEM_ERR 1

enum ENUM_STAT_ITEM
{
STAT_ITEM0,
STAT_ITEM1,
STAT_ITEM_BUTT
};

typedef struct tag_PERM_COUNT_STAT_INFO(
unsigned short stat_item ;
unsigned short number;
} _SPermCountStatInfo;

```

```

_SPermCountStatInfo pcsi[STAT_ITEM_BUTT] =(
( STAT_ITEM0, 16000},
( STAT_ITEM1, 50000} ,}

unsigned long *pulStatDataBuffer = NULL;unsigned short

AllocPermMemory(void) {

unsigned short usBufferSize = 0;

unsigned short usLoop = 0;

for(usLoop = 0; usLoop < STAT_ITEM_BUTT; usLoop ++)

{

usBufferSize += pcsi[usLoop].number ;

}

pulStatDataBuffer = (unsigned long*)malloc(sizeof(unsigned long) * usBufferSize);

if (NULL == pulStatDataBuffer)

{

return MEM_ERR;

}

return MEM_OK;

}

```

下面语法正的是;,, 确

A unsigned short 语型不能表示循语中体将 16000 与 50000 相加的和 66000

B 语数 pcsi的 number 域是 unsigned short语型 不能表示 50000 语语大的

字数 C 循语件语语改语条 usLoop<=STAT_ITEM_BUTT2ig 指出下面语段代语中的语
语,

```

unsigned long FuncB ( unsigned long ulCount )

```

```

{
unsigned long ulSum = 0 ;
while( 0 <= ulCount )
{
ulSum += ulCount ;
ulCount--;
}
return ulSum ;
}

void test(void)
{
unsigned long ulTotal = 0;
ulTotal=FuncB(10);

printf("%lu" ulTotal);}

```

下面描述正确的是(),

A、while 循环语句判断语句 断真

B、 test 打印输出 55

C、 循环在语句体内 2 的 32 次方后 ulSum 语句始溢出语 翻

22. 指出下面程序的语句

```

void Test(void)
(
char *szStr = (char *) malloc(100);

if(NULL == szStr)

```



```

(
return;
}
strcpy(szStr, "hello");
free(szStr);
if(NULL != szStr)
(
strcpy(szStr, "world");
printf("%s", szStr);
}
return;
}

```

下面描述正确的是： ， 确

- A、 strcpy有语尾符拷语到没将 szStr中
- B、 语语放空语的指语语行拷语操作
- C、 szStr被 free后 szStr语空。即

23. #define BUFFER_SIZE 256void GetMemory(char **ppszBuf)

```

(
if(NULL == ppszBuf)
(
ASSERT(0);
return ;
}
*ppszBuf = (char *)malloc(BUFFER_SIZE);
return;
}

```

```

void Test(void)
(
char *pszBuf=NULL;
GetMemory(&pszBuf);
strcpy(pszBuf, "hello world\r\n");
printf("%s", pszBuf);
free(pszBuf);
return;
}

```

下面语法正的是;,, 确

A、 pszBuf 的语永语语 NULL^

B、 malloc 存后有判是否成功 内没断

C、 strcpy拷语越界

DK GetMemory 无法申语的存地址语语语将内 pszBuf

24、 语语下面函中数 1、 2、 3 语语入什语语句才合理, ;填

A、 不添, 不添 不添

B、 free(pMsg); , free(ptmpMsg)不添 C、
free(pMsg);, free(ptmpMsg);, free(ptmpMsg);

D 不添, free(pMsg);, free(ptmpMsg);E
free(pMsg); , free(pMsg); , free(ptmpMsg);

F、 不添, 不添, free(ptmpMsg);

```
#define COMM_MSG_LEN 100
```

```
char *GetMessageBuffer(int malloc_len) (
```

```
char *ptr=NULL;
```

```
ptr=(char *)malloc(malloc_len)
```

```
return ptr;
```

```
}
```

```
int FuncTest(void)
```

```
(
```

```
int malloc_len = COMM_MSG_LEN;
```

```
char *pMsg=NULL;
```

```
char *ptmpMsg=NULL;
```

```
pMsg = GetMessageBuffer(malloc_len);
```

```
if(NULL == pMsg)
```

```
(
```

```
___ 1___
```

```
return ERROR;
```

```
}
```

```
fillMessage(pMsg);
```

```
sendMessage(pMsg);
```

```

ptmpMsg = GetMessageBuffer(malloc_len);

if(NULL == ptmpMsg)

{
    __ 2__
return ERROR;
}

FillMessage(ptmpMsg);

SendMessage(ptmpMsg);

free(pMsg);

__ 3__

return OK;
}

```

25. 以下述中不正的是叙确()

- A) 在不同的函中可以使用相同名字的量数
- B) 函中的形式是在语中保存数参数
- C) 在一函定语的量数只在本函范语有效个数内数内
- D) 在一函的语合语句中定语的量数在本函范语有效个数内数内 (语合语句指函

中的成语括成数号构的代语)

26. 全局量数可以定语在被多个.C文件包含着的语文件中。()

- A. 正确 B. 语语

27. 在函部定语的量数内(语量、寄存器量等特殊量数静除外)的存是在 语存中 内内所以在定语函数内内部的量的语候 一定要保语语不能语溢出。如 果语语量占用空语语大 语语使用存申语的方式 语语语量指向的存内内就是 在堆存中

了。()

A. 正确 B. 错误

28. 局部变量可以和全局变量重名 编译的时候不出错 会但一旦不小心 就可能导致使用全局变量 所以在定义局部变量的时候 不要和全局变量重名 C ()

A. 正确 B. 错误

29. 程序有如下定义:

```
unsigned long pulArray[] = {6, 7, 8, 9, 10};  
unsigned long *pulPtr;
```

程序下列程序段的输出结果是()

```
pulPtr = pulArray;  
*(pulPtr + 2) += 2;  
printf ("%d,%d\n", *pulPtr, *(pulPtr + 2));
```

A) 8, 10 B) 6, 8 C) 7, 9 D) 6, 10

30. struct stu

```
{  
    int num;  
    char name[10];  
    int age;  
};  
void Fun(struct stu *p)  
{  
    printf("%s\n", (*p).name);  
}
```

```
return;
```

```

void main()
(
struct stu students[3]={ {9801, "Zhang", 20},
{9802, "Wang", 19},
{9803, "Zhao", 18} };
Fun(students + 2);
return;
}

```

语出语果是()

A) Zhang B) Zhao C) Wang D) 18

31. 以下程序行后运，语出语果是()

```

void main( )
{
char *szStr = "abcde";
szStr += 2;
printf("%lu \n", szStr);
return;
}

```

A cde B 字符 c 的 ASCLL® 语

C "abcde"语常申中字符个 c 所在的地址 D 出语 32. 语有以下宏定语:

```
#define N 3
```

```
#define Y(n) ((N+1)*n) /* 语语定语在语程语范中是不允语的 如果在语语
```

中语语用了你

```
语自打手心三下*/
```

语句行语句: $z = 2 * (N + Y(5 + 1))$ z的语句()

A) 出语 B) 42 C) 48 D) 54

33、有如下宏定语和语定语构

```
#define MAX_SIZE A+B

struct _Record_Struct
{
unsigned char Env_Alarm_ID :4;
unsigned char para1 :2;
unsigned char state;
unsigned char avail:1;
} * Env_Alarm_Record ;

pointer = (struct _Record_Struct *)malloc(
sizeof(struct _Record_Struct) * MAX_SIZE);
```

当 A=2, B=3 语 pointer分配()字语的空语。 个注意, 机器是按照 4 字语语语

A) 20 B) 15 C) 11 D) 934、语语下面语目中 a, b, c语出语语多少, ; ,

A、 10, 210, 30

B、 100, 210, 300

C、 110, 210, 310

» 100, 210, 30

E、 110, 200, 30

```
#include <stdio.h>
```

```
#define FUNC(A) A++;\
```

```
A *= 10;int main(int argc, char* argv[]) {
```

```
int a = 10;
```



```
int b = 20;

int c = 30;

if(a > 10)
FUNC(a);

if(b >= 20)
FUNC(b);

if(c > 30)
(
FUNC(c);
}

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

return 0;

}
```

35. 在 X86 下, 有下列程序

```
#include <stdio.h>

void main()

(
union

(
int k;
char i[2];
}*s, a;

s = &a;

s->i[0] = 0x39;

printf("%x\n", a.k);

}
```

语出语果是()

A) 3839 B) 3938 C) 380039 D) 不可语知

36、#define MAX_MEM_SIZE 100int GetMem(int iLen, void **ppMem)

```
{
if(NULL==ppMem)
{
return ,1;
}
if(iLen<=0)
{
return 0;
}
else if(iLen<MAX_MEM_SIZE)
{
*ppMem=malloc(iLen);
return iLen;
}
else
{
*ppMem=malloc(MAX_MEM_SIZE);
return MAX_MEM_SIZE;
}
void main()
(
char *pMyMem;
int i;
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/598056063106006103>