部门 _____ 科室 _____ 姓名 _____ 工号 _____

1 单选题 (每题3分,共计36分)

1.1 有以下程序

```
main()
{
    int a,b,d=25;
    a=d/10%9;b=a&&(-1);
    printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是 _____

A) 6,1　　B) 2,1

C) 6,0　　D) 2,0

1.2 有以下程序

```
main()
{
    char a[7]="a0\0a0\0"; int i,j;
    i=sizeof(a);
    j=strlen(a);
    printf("%d %d\n",i,j);
}
```

程序运行后的输出结果是 _____

A) 2 2　　B) 7 6

C) 7 2　　D) 6 2

1.3 有以下程序

```
main()
{
    int a[3][3],*p,i;
    p=&a[0][0];
    for(i=0;i<9;i++)p[i]=i;
    for(i=0;i<3;i++)printf("%d",a[1][i]);
}
```

程序运行后的输出结果是 _____

A) 0 1 2　　B) 1 2 3

C) 2 3 4　　D) 3 4 5

1.4 有以下程序

```
#define N 20
fun(int a[],int n,int m)
{
    int i,j;
    for(i=m;i>=n;i--)a[i+1]=a[i];
}
main()
{
    int i,a[N]={1,2,3,4,5,6,7,8,9,10};
    fun(a,2,9);
    for(i=0;i<5;i++)printf("%d",a[i]);
}
```

程序运行后的输出结果是 _____

A) 10234　　B) 12344

C) 12334　　D) 12234

1.5 有以下程序

```
main()
{
    int a[3][2]={0},(*ptr)[2],i,j;
    for(i=0;i<2;i++)
    {
        ptr=a+i;scanf("%d",ptr);ptr++;
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%2d",a[i][j]);
        }
        printf("\n");
    }
}
```

若运行时输入:1 2 3 回车,则输出结果是 _____

A) 产生错误信息

B) 1 0
2 0
0 0

C) 1 2
3 0
0 0

D) 1 0
2 0
3 0

1.6 有以下程序

```
prt(int *m,int n)
{
    int i;
    for(i=0;i<n;i++) *m++ = *m+1;
}
main()
{
    int a[]={1,2,3,4,5},i;
    prt(a,5);
    for(i=0;i<5;i++)
        printf("%d,",a[i]);
}
```

程序运行后的输出结果是 _____

A) 1,2,3,4,5,　　B) 2,3,4,5,6,

C) 3,4,5,6,7,　　D) 2,3,4,5,1,

1.7 有以下程序

```
#define P 3
void F(int x)
{
    return(P*x*x);
}
main()
{
    printf("%d\n",F(3+5));
}
```

程序运行后的输出结果是 _____

A) 192　　B) 29

C) 25　　D) 编译出错

1.8 有以下程序

```
main()
{
    int c=35;
    printf("%d\n",c&c);
}
```

程序运行后的输出结果是 _____

A) 0　　B) 70

C) 35　　D) 1

1.9 有以下程序

```
main()
{
    int a=1,b;
    for(b=1;b<=10;b++)
    {
        if(a>=8)
        break;
        if(a%2==1)
        {
            a+=5;
            continue;
        }
        a-=3;
    }
    printf("%d\n",b);
}
```

程序运行后的输出结果是 _____

A) 3　　B) 4

C) 5　　D) 6

1.10 有以下程序

```
main()
{
    char s[]="159",*p;
    p=s;
    printf("%c",*p++);
    printf("%c",*p++);
}
```

程序运行后的输出结果是 _____

A) 15　　B) 16

C) 12　　D) 59

1.11 有以下程序

```
main()
{
```

```
    int num[4][4]={{1,2,3,4},{5,6,7,8},
                   {9,10,11,12},{13,14,15,16}};
    int i,j;
    for(i=0;i<4;i++)
    {
        for(j=0;j<=i;j++)printf("%4c",' ');
            for(j=_____;j<4;j++)printf("%4d",num[i][j]);
                printf("\n");
    }
}
```

若要按以下形式输出数组右上半三角

```
1 2 3 4
6 7 8
11 12
16
```

则在程序下划线处应填入的是_____

A) i-1　　B) i

C) i+1　　D) 4-i

1.12 有以下程序

```
void point(char *p)
{
    p+=3;
}
main()
{
    char b[4]={'a','b','c','d'},*p=b;
    point(p);
    printf("%c\n",*p);
}
```

程序运行后的输出结果是_____

A) a　　B) b

C) c　　D) d

2 填空题 (每题3分,共计18分)

2.1 以下程序运行后的输出结果是_____

```
#define S(x) 4*x*x+1
main()
{
    int i=6,j=8;
    printf("%d\n",S(i+j));
}
```

2.2 以下程序运行后的输出结果是_____

```
main()
{
    int a,b,c;
    a=10;b=20;c=(a%b<1)||(a/b>1);
    printf("%d %d %d\n",a,b,c);
}
```

2.3 以下程序运行后的输出结果是_____

```
fun(int a)
{
    int b=0;static int c=3;
    b++;c++;
    return(a+b+c);
}
main()
{
    int i,a=5;
    for(i=0;i<3;i++)
        printf("%d %d ",i,fun(a));
    printf("\n");
}
```

2.4 以下程序运行后的输出结果是_____

```
struct NODE
{
    int k;
    struct NODE *link;
};
main()
{
    struct NODE m[5],*p=m,*q=m+4;
    int i=0;
    while(p!=q)
    {
        p->k=++i;p++;
        q->k=i++;q--;
    }
    q->k=i;
    for(i=0;i<5;i++)
        printf("%d",m[i].k);
```

```
    printf(”\n”);
}
```

2.5 以下程序运行后的输出结果是 _____

```
#include ”stdio.h”
int main()
{
    int a;
    int *p;
    p = &a;
    *p = 0x500;
    a = (int )(*(&p));
    a = (int )(&(*p));
    if(a == (int)p)
        printf(”equal !\n”);
    else
        printf(”not equal !\n”);
}
```

2.6 以下程序的输出显示

```
unsigned int x = 0x123;
unsigned char *puch = (unsigned char *)&x;

printf(”%x,%x,%x,%x\n”,  puch[0],puch[1],puch[2],puch[3]);
```

BIG-ENDIAN 情况下输出: _____

LITTLE-ENDIAN情况下输出: _____

3 改错题 (共计26分)

3.1 请找出下面代码中的隐患或者错误 (5分)

```
BOOL H248_GetPtFrmStr(H248_SDP_FORMAT  tSdpFmt,BYTE * pbVal)
{
    BYTE bFmtStr[H248_MAX_FMT_STRING_LEN] ;

    if((H248_UNUSE ==tSdpFmt.bUseFlag)||(NULL == pbVal))
    {
        return H248_BL8_FALSE;
    }
    H248_strncpy(bFmtStr,tSdpFmt.bStrContent,H248_MAX_FMT_STRING_LEN,
       tSdpFmt.bStrLen);
    H248_StrTurnToUpper(bFmtStr);
    if(gdwdebug == 7 )
    {
        return H248_BL8_FALSE;
    }
    if(H248_StrCmp(bFmtStr,”G721”))
    {
        *pbVal = PACK_TYPE_G721 ;
        return H248_BL8_TRUE;
    }
    else if(H248_StrCmp(bFmtStr,”G722”))
    {
        *pbVal =  PACK_TYPE_G722;
        return H248_BL8_TRUE;
    }
    return H248_BL8_FALSE;
}
void H248_StrTurnToUpper(char *str)
{
    if(str == NULL)
    {
        return;
    }

    while( (*str) != 0)
    {
        if( (*str) > 0x60 && (*str) < 0x7B)
        {
            (*str) = (*str) - 0x20;
        }
        str++;
    }
}
BOOL H248_strncpy(char *pDest, char *pSrc, DWORD dest_Len, DWORD
   copy_Len)
{
    if((pDest == NULL)||(pSrc == NULL))
    {
        H248_INCR_VAL(gvH248_Statis.tError.tTools.dwStrCpy,
           H248_COUNT_STEP);
        return FALSE;
    }
    if((copy_Len >= dest_Len)||(dest_Len <= 0)||(copy_Len <= 0))
    {
        H248_INCR_VAL(gvH248_Statis.tError.tTools.dwStrCpy,
           H248_COUNT_STEP);
        return FALSE;
    }
    memcpy((void *)pDest, (void *)pSrc, (size_t)copy_Len);
    *(pDest+copy_Len) = ’\0’;
    return TRUE;
}
```

3.2 请找出下面代码中的隐患或者错误 (5分)

```
WORD32 CSCF_REG_PrintInfoToOMC(BYTE *pbData,WORD32 *pdwDataLen,
                               T_CSCF_MML_COMMANDG *ptMMLCommand)
{
    WORD32 j=0;
    JID    tRegJid={0};
    WORD32 dwRetCode = CMS_SUCCESS;
    WORD16 wCurStat  = 0;

    if ((pbData==NULL)||(pdwDataLen==NULL)||(ptMMLCommand==NULL))
    {
        return CSCF_MML_FAILURE;
    }
    if (CMS_SUCCESS!=CSC_XOS_GetJID(JOBTYPE_CSF_REG, 1, &tRegJid))
    {
        j+=snprintf((CHAR *)pbData,
                    128,
                    "CSC_REG OSS_GetPIDByName Error,\\
OMC COMMAND FAIL\n");
        pbData[j]='\0';
        *pdwDataLen=strlen((LPSTR)pbData);
        return CSCF_MML_FAILURE;
    }
    if( CMS_SUCCESS!=CSC_XOS_GetCurState(&wCurStat) )
    {
        /* reg 进程如果不等于 WORK/SLAVE 状态,直接返回进程未起来 */
        j+=snprintf((CHAR *)pbData,
                    128,
                    "CSC_REG CSC_XOS_GetCurState Error,\\
OMC COMMAND FAIL\n");
        pbData[j]='\0';
        *pdwDataLen=strlen((LPSTR)pbData);
        return CSCF_MML_FAILURE;
    }
    ......
    return CSCF_MML_SUCCESS;
}
```

3.3 请找出下面代码中的隐患或者错误 (5分)

```
#define SIP_FEATURES_SUPPORT_PRACK          0x00000001
#define SIP_FEATURES_SUPPORT_UPDATE         0x00000002
#define SIP_FEATURES_SUPPORT_EARLYSESSION   0x00000010
#define SIP_FEATURES_SUPPORT_PRECONDITION   0x00000020
#define SIP_FEATURES_REQUIRE_PRACK          0x00010000
#define SIP_FEATURES_REQUIRE_EARLYSESSION   0x00100000
#define SIP_FEATURES_REQUIRE_PRECONDITION   0x00200000

void C_IMSBCM_FillStimulatedFeatures(SIP_FEATURES* ptDestFeatures,
                                     SIP_FEATURES* ptSourceFeatures,
                                     BOOL blActiveState)
{
    memcpy(ptDestFeatures, ptSourceFeatures, sizeof(SIP_FEATURES));
    if (ptDestFeatures->dwFeatures &
        SIP_FEATURES_REQUIRE_PRECONDITION)
    {
        ptDestFeatures->dwFeatures &= !
            SIP_FEATURES_REQUIRE_PRECONDITION;
        ptDestFeatures->dwFeatures |=
            SIP_FEATURES_SUPPORT_PRECONDITION;
    }
    if (ptDestFeatures->dwFeatures & SIP_FEATURES_REQUIRE_PRACK)
    {
        ptDestFeatures->dwFeatures &= !SIP_FEATURES_REQUIRE_PRACK;
    }
    ptDestFeatures->dwFeatures |= SIP_FEATURES_SUPPORT_PRACK;
    if (TRUE_B8 == blActiveState)
    {
        ptDestFeatures->dwFeatures =
            SIP_FEATURES_SUPPORT_EARLYSESSION;
    }
    ptDestFeatures->bType |= SIP_FEATURES_CREATE;
    return;
}
```

3.4 请找出下面代码中的隐患或者错误 (5分)

```
BOOL8  D_DR_AlarmSendComInfo(WORD wAlarmCode,void * pAlarmInfo,BYTE
    bLength)
{
    if(NULL == pAlarmInfo)
    {
        bLength = 0;
    }
    D_DR_ALOCK();  /* 加锁 */
    if(tDRAlarmPool.wCount)
    {
        /* 先发送告警池的告警 */
        if(!D_DR_SendAlarmInPool())
        {
            /* 把最新的告警加入告警池 */
            D_DR_ForceAddAlarm2Pool(wAlarmCode,pAlarmInfo,bLength);
            return FALSE;
        }
    }
    if(!D_DR_BaseSendAlarm(wAlarmCode,pAlarmInfo,bLength))
    {
        D_DR_ForceAddAlarm2Pool(wAlarmCode,pAlarmInfo,bLength);
```

```
    }
    D_DR_AUNLOCK();/* 解锁 */
    return TRUE;
}
```

3.5 请找出下面代码中的隐患或者错误 (6分)

```
#define DB_SUBSYS_H248  0xff1c

DBBOOL  _func_get_all_tuplehandle_by_ipaddr(
                      _db_t_database_handle dbHandle,
                      _db_t_table_handle    tableHandle,
                      T_IPAddr              *pIPAddr,
                      LP_T_DB_HANDLE_LIST   ptHandleList)
{
    _DB_RET  bret,ret;
    T_IPAddr tSubNet = {0};
    T_IPAddr tMask = {0};
    WORD     wLength=0;
    BYTE     IpLength =0;
    DWORD    i, j;
    BYTE     bMskSize =0;
    BYTE     abMaskSizeList[ DB_TMPRESULT_M]  = {0};
    BYTE     bEqualFlag=0;

    if (pIPAddr == NULL || ptHandleList == NULL)
    {
        XOS_SysLog(DBS_PRNLEVEL_ERROR,
                   "[PSSDB]:Line:%d
                       _func_get_all_tuplehandle_by_ipaddr\\
( :%s failed \n",
                   __LINE__,
                   DB_SUBSYS_H248 );
        return FALSE;
    }
    ptHandleList->dwHandleNum = 0;
    /* 根据入参的数据库和表句柄,遍历该表 */
    bret=_db_skip_first_tuple(dbHandle,tableHandle, &htp_handle);
    while ((bret== _DB_FOUND) && (ptHandleList->dwHandleNum <
       DB_TMPRESULT_M) )
    {
        .......
        bEqualFlag = TRUE;
        /* ipv4 取四个字节,取个字节,IPV616分别与
            Mask 相与,如果等于子网, IP记录下掩码的为
            1 的个数,与当前的掩码的计数器比较,直到找到最长匹配的记录
            */
        for (i=0; i< IpLength;i++)
        {
            /* IP 地址与掩码相与后不等于子网,将标志置为
                FALSE 不取当前记录, */
            if((pIPAddr->unIPAddrValue.
                    byIPV6Addr[i]&tMask.unIPAddrValue.byIPV6Addr[i])
                       !=
               (tSubNet.unIPAddrValue.
                    byIPV6Addr[i]&tMask.unIPAddrValue.byIPV6Addr[i]))
            {
                bEqualFlag = FALSE;
                break;
            }
        }
        if ( bEqualFlag == TRUE) /* IP 地址与掩码相与后等于子网的情况 */
        {
            ret=bCountOneBits(tMask.unIPAddrValue.byIPV6Addr,
               CSCF_IPV6ADDR_LEN,
                           &bMskSize);
            if (ret)
            {
                for (i = 0; i < ptHandleList->dwHandleNum; i++)
                {
                    if (bMskSize > abMaskSizeList[i])
                    {
                        /* 从 i 开始的数据后移一位 */
                        for (j = ptHandleList->dwHandleNum-1; j >= i
                                    ; j--)
                        {
                            abMaskSizeList[j+1]            =
                               abMaskSizeList[j];
                            ptHandleList->atTupleHandle[j+1] =
                               ptHandleList->atTupleHandle[j];
                        }
                        break;
                    }
                }
                /* 将新数据插入数组中 */
                abMaskSizeList[i] = bMskSize;
                ptHandleList->atTupleHandle[i] = htp_handle;
                ptHandleList->dwHandleNum++;
            }
            bMskSize =0;
        }
        bret = _db_skip_next_tuple(dbHandle,tableHandle,
                                   htp_handle,&htp_handle);
    }
    ......
}
```

4　编程题 (共计20分)

设计一个算法,判断一个算术表达式中的括号是否配对,算术表达式保存字符串中。在进行检测括号是否匹配的时候,需要考虑到各种情况:

(1) 匹配。例如:(())
(2) 左括号不匹配。例如:(()
(3) 右括号不匹配。例如:())
(4) 其他情况,例如:)(

函数原型:`int MatchBracket(const char *chExpr)`

参数说明:`chExpr`表示待分析的算术表达式字符串

返回值说明:返回匹配结果,能够根据返回值区分各种不匹配情形。