C

C

1

"::" ;

2

extern extern

extern

3 .C

C static C

C .

4

```
#include <stdio.h>
int main(void)
{
    int a,b,c,d;
    a=10;
    b=a++;
    c=++a;
   d=10*a++;
    printf("b c d %d %d %d" b c d ;
    return 0;
  }
    10 12 120
```

5 static static static

: 1) ( ) static

2)

3) static

, (static)

:

static

static ;

```
   static
        static
   static
        static
```

6

7

```
   typedef union
  {
   long i;
   int k[5];
  char c;
  } DATE;
 struct data
 {
 int cat;
 DATE cow;
   double dog;
  } too;
 DATE max;
      printf("%d",sizeof(struct data)+sizeof(max));
```

___52____

: struct union.(32)

DATE union, . int[5], 20 .

20. data struct, . int4 + DATE20 + double8 = 32.

20 + 32 = 52. ... 16 , int 2 , int2 + DATE10 + double8 = 20

8

9

```
 #include <stdio.h>
 int inc(int a)
 { return(++a); }
int multi(int*a,int*b,int*c)
{    return(*c=*a**b); }
 typedef int(FUNC1)(int in);
typedef int(FUNC2) (int*,int*,int*);
   void show(FUNC2 fun,int arg1, int*arg2)
    {
 FUNC1 p=&inc;
   int temp =p(arg1);
   fun(&temp,&arg1, arg2);
   printf("%dn",*arg2);
 }
 main()
 {
   int a;          //       a   0;
   show(multi,10,&a);
   return 0;
     }
      110
```

10 (,)

“abcd” “dcba”

```
#include"string.h"
main()
{
char*src="hello,world";
char* dest=NULL;
int len=strlen(src);
dest=(char*)malloc(len);
char* d=dest;
char* s=src[len];
while(len--!=0)
d++=s--;
printf("%s",dest);
return 0;
}
```

1 4 ;

```
int main()
{
  char* src = "hello,world";
  int len = strlen(src);
  char* dest = (char*)malloc(len+1);//                    char* d = dest;
  char* s = &src[len-1];        //
  while( len-- != 0 )
   *d++=*s--;
  *d = 0;          //          '\0'
  printf("%sn",dest);
  free(dest);       //
  dest = NULL;      //
  return 0;
  }
```

2 (, .)

```
#include <stdio.h>
#include <string.h>
main()
{
char str[]="hello,world";
int len=strlen(str);
char t;
for(int i=0; i<len/2; i++)
{
t=str[i];
str[i]=str[len-i-1]; //
str[len-i-1]=t;
}
printf("%s",str);
return 0;
}
```

11. , C , C++ ?

c c++ inline

12. ?

PPP

13. ?

V5

14.voip ?

H.323 SIP Skype H.248 MGCP

15. ?

16. ?

17.

```
unsigned char *p1;
unsigned long *p2;
p1=(unsigned char *)0x801000;
p2=(unsigned long *)0x810000;
    p1+5= ;
     p2+5= ;
```

:0x801005(5) 0x810014(20);

:

21.Ethternet Internet ? D

A.HDLC;B.ARP;C.UDP;D.TCP;E.ID

22. :(B C)

A.TCP;B.IP;C.ICMP;D.X.25

23.Windows :(C)

A. ;B. ;C. ;D. ;

:

25. ?

```
int a[60][250][1000],i,j,k;
for(k=0;kMax_GT_Length)
{          return GT_Length_ERROR;
  }         ....... }
```

: / /

:

29.IP Phone ?

IP IP PHONE VoIP IP ,

, IP , IP

, , , , IP

30.TCP/IP

31.1 7

1

7

32. 5

“ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ” “ ”

. :

1. ?

```
int a[60][250][1000],i,j,k;
for(k=0;k<=1000;k++)
for(j=0;j<250;j++)
for(i=0;i<60;i++)
a[i][j][k]=0;
```

:

```
2.#define Max_CB 500
 void LmiQueryCSmd(Struct MSgCB * pmsg)
 {
  unsigned char ucCmdNum;
 ......
   for(ucCmdNum=0;ucCmdNum<Max_CB;ucCmdNum++)
  {
  ......;
  }
```

: ,unsigned int 0~255

3. , :

```
 #define SQUARE(a)((a)*(a))
 int a=5;
 int b;
 b=SQUARE(a++);
```

: , .

1

: , .

:

(1)

(2)

(3) .

(4)

2.

:

3 Heap stack

:Heap stack

Stack / Heap /
Stack Heap
C malloc ,C++ new
,

4 Windows

8 IA32
(4K) (4M)
9

10 IA32
ring3 ring0 ring3 ring0 vm86
11 winamp
..
FindWindow , .
12 'a' 'b'
: SetWindowsHookEx
14
: sql
SQL

15 Template
: Template , .
, Template.
STL , , .
(C++ template
template template)
16 Windows DNA
:Windows Distributed interNet Application Architecture Windows
Windows DNA
Windows DNA Internet / PC

17.
:1)
2)

SMP

1. swap x y

```
#define swap(x, y)
x = x + y;
y = x - y;
x = x - y;
```

2. a[N] 1 N-1 .
o N

int do_dup(int a[],int N)

:int do_dup(int a[],int N)   / /

```
{
    int sun = 0;
    int sum2;
    for(int i=0;i<N;++i)
    {
     Sum+=a[i];
    }
    Sum2 = (1+N-1)*N/2;
    Return (sum-sum2);
}
```

3 x 2

: 1)int i = 512;

```
cout << boolalpha << ((i & (i - 1)) ? false : true) << endl; / /        0,          2
```

2)return (x>>N==1);

4.unsigned int intvert(unsigned int x,int p,int n) x ,p ,n

, . x=0b0001 0001,p=4,n=3 x=0b0110 0001

:unsigned int intvert(unsigned int x,int p,int n) / / p=4,n=3

```
{
  unsigned int _t = 0;
  unsigned int _a = 1;
   for(int i = 0; i < n; ++i)/ /                  -t
   {
    _t |= _a;      / /
    _a = _a << 1;
    }
   _t = _t << p;   / /       _t      1110000
   x ^= _t;      /     ,
   return x;
}
```

1.

: , # .

?

)

)

2.

```
char * const p;
char const * p
const char *p

char * const p; / /            p
char const * p   / /
const char *p    / /    char const *p
```

3.

```
char str1[] = "abc";
char str2[] = "abc";
const char str3[] = "abc";
const char str4[] = "abc";
const char *str5 = "abc";
const char *str6 = "abc";
char *str7 = "abc";
char *str8 = "abc";
cout << ( str1 == str2 ) << endl;
cout << ( str3 == str4 ) << endl;
cout << ( str5 == str6 ) << endl;
cout << ( str7 == str8 ) << endl;
```

0 0 1 1

str1,str2,str3,str4

str5,str6,str7,str8

4. sizeof [C]

```
void UpperCase( char str[] ) //  str
{
for( size_t i=0; i<sizeof(str)/sizeof(str[0]); ++i )
if( 'a'<=str[i] && str[i]<='z' )
str[i] -= ('a'-'A' );
}
char str[] = "aBcDe";
cout << "str : " << sizeof(str)/sizeof(str[0]) << endl;
UpperCase( str );
cout << str << endl;
```

sizeof sizeof

str 6 str

sizeof 4

4

: , .

sizeof() , , sizeof .

4. 32 ,

80386 32 4

5. (,)

```
main()
{
int a[5]={1,2,3,4,5};
int *ptr=(int *)(&a+1);
printf("%d,%d",*(a+1),*(ptr-1));
}
```

2,5

*(a+1 a[1] *(ptr-1) a[4], 2 5

&a+1 +1 a 5 int

int *ptr=(int *)(&a+1);

ptr &(a[5]), a+5

&a int (*)[5];

1

+1

a 5 int 5*sizeof(int)

ptr a[5]

prt (&a+1) ( )

prt-1 sizeof(int*)

a,&a a a[0] &a a+1

a[1],&a+1 a[5].

6.

1).

```
int main()
{
char a;
char *str=&a;
strcpy(str,"hello");
printf(str);
return 0;
}
```

; str

Strcpy string.h . //

2).

```
char* s="AAA";
printf("%s",s);
s[0]='B';
printf("%s",s);
```

: "AAA" s s

cosnt char* s="AAA";

s[0]

1 “ ”

:#define Min(X, Y) ((X)>(Y)?(Y):(X)) // ;

2 C

:while(1){} for(;;) //

3 static

:1) , .

static

3) static

4 const

:1)

2) , .

3) , .

4) , .

5 volatile

:

: ;

6. int (*s[10])(int)

int (*s[10])(int) int func(int param)

1.

```
int a=248; b=4;int const c=21;const int *d=&a;
int *const e=&b;int const *f const =&a;
```

:*c=32;d=&b;*d=43;e=34;e=&a;f=0x321f;

*c

*d const

e = &a const

const *f const =&a;

2. a=3,b=5, a=5,b=3;

: , , ^( )

```
 a = a + b;
 b = a - b;
 a = a - b;
or
 a = a^b;//  int,char..
 b = a^b;
 a = a^b;
or
 a ^= b ^= a;
```

3.c c++ struct

:c c++ struct c struct c++ struct c++ struct class struct public class private.

```
4.#include <stdio.h>
 #include <stdlib.h>
 void getmemory(char *p)
{
 p=(char *) malloc(100);
}
int main( )
{
char *str=NULL;
getmemory(str);
strcpy(p,"hello world");
printf("%s/n",str);
free(str);
return 0;
}
```

: getmemory malloc free str

5.char szstr[10];

strcpy(szstr,"0123456789");

; , OS, .

6.

:

7.

silver6 | 02 , 2007 11:41

Author Vince

1

1.1

1. const const char*, char const*, char*const C++ Bjarne The C++ Programming Language

char * const cp; (* pointer to)

cp is a const pointer to char

const char * p;

p is a pointer to const char;

char const * p;

C++ const* const

2. c

int *p[n];-----

int (*p)[n];------p n

int *p();----------

int (*p)();------p

3. ( )

:

```
#define MAX 255
int main()
{
unsigned char A[MAX],i;
for (i=0;i<=MAX;i++)
A[i]=i;
}
```

MAX=255, A :0..MAX-1, , i 255 , : A[255]=255; for (i=0;i<=MAX;i++) , unsigned char (0..255),i++ i 0 .. .

char [-128 127] unsigned char [0 ,255]

4. C++:memset ,memcpy strcpy

:#include "memory.h"

memset ' ' " :char a[100];memset(a, '', sizeof(a));

memcpy char a[100],b[50]; memcpy(b, a, sizeof(b)); sizeof(a) b

strcpy '\0' char a[100],b[50];strcpy(a,b); strcpy(b,a) a '\0' 50 b

strcpy

```
    extern char *strcpy(char *dest,char *src);
  {
    ASSERT((dest!=NULL)&&(src!=NULL));
    Char *address = dest;
    While((*dest++=*src++)!='\0')
      Continue;
    Return dest;
  }
```

#include <string.h>

src NULL dest

src dest dest src

dest

memcpy

```
    extern void *memcpy(void *dest, void *src, unsigned int count);
  {
   ASSERT((dest!=NULL)&&(src!=NULL));
   ASSERT((dest>src+count)||(src>dest+count));//            ,        restrict
   Byte* bdest = (Byte*)dest;
   Byte* bsrc = (Byte*) src;
   While(count-->0)
     *bdest++ = **bsrc++;
   Return dest;
  }
```

#include <memory.h>

src count dest

src dest dest

Memset

```
    extern void *memset(void *buffer, char c, int count);
```

#include

buffer count c

buffer

5. ASSERT()

:ASSERT() FALSE (0), 0

n

0 0

```
......
ASSERT( n != 0);
k = 10 / n;
......
```

ASSERT Debug Release

assert() ANSI C ASSERT Release

6. system "pause"); ?

: " " getchar

7. C++ C struct

:c++ oo c struct

c struct , , .

8.

:

: ,override, overload( ),

, , .. , , ,

. .

, , //

9.

;

main

main

10. 8086

:8086 16 20

12

1. C n

```
:long int fact(int n)
{
 If(n==0||n==1)
   Return 1;
 Else
  Return n*fact(n-1);
}
```

2.

1)

```
int BSearch(elemtype a[],elemtype x,int low,int high)
/*      low      high     a           x*/
{
```

```
int mid;
if(low>high) return -1;
mid=(low+high)/2;
if(x==a[mid]) return mid;
if(x<a[mid]) return(BSearch(a,x,low,mid-1));
else return(BSearch(a,x,mid+1,high));
}
2)
int BSearch(elemtype a[],keytype key,int n)
{
int low,high,mid;
low=0;high=n-1;
while(low<=high)
{
mid=(low+high)/2;
if(a[mid].key==key) return mid;
else if(a[mid].key<key) low=mid+1;
else high=mid-1;
}
return -1;
}
3.
f(1)=1
f(2)=1
f(n)=f(n-1)+f(n-2) n>2
                  :
int f(int n)
{
int i,s,s1,s2;
s1=1;/*s1           f(n-1)       */
s2=1;/*s2           f(n-2)       */
s=1;
for(i=3;i<=n;i++)
{
s=s1+s2;
s2=s1;
s1=s;
}
return(s);
}
          :
Int f(int n)
{
 If(n==1||n==2)
   Rerurn 1;
 Else
   Rerutn f(n-1)+f(n-2);
}
```

4.

```
  :int a = ……;
  int b = ……;
  a = a + b;
  b = a - b;
  a = a - b;
```

5.

```
  :void BubbleSort(elemtype x[],int n)   //              0(n*n);
 {
  int i,j;
  elemtype temp;
  for(i=1;i<n;i++)
   for(j=0;j<n-i;j++)
  {
   if(x[j].key>x[j+1].key)
     {
     temp=x[j];
     x[j]=x[j+1];
     x[j+1]=temp;
     }
  }
}
//                          :
 void BubbleSort(elemtype x[],int n)
{
 Int i,j;
 BOOL exchange;  //
```

```
{
FILE *fp;
char ch,filename[10];
scanf("%s",filename);
if((fp=fopen(filename,"w")==NULL)
{
 printf("cann't open filen");
 exit(0);
}
ch=getchar();
while(ch!='#')
{
fputc(ch,fp);
putchar(ch);
ch=getchar();
}
fclose(fp);
}
```

7. winsocket //

```
//
#include <Winsock2.h>
#include <stdio.h>
void main()
{
WORD wVersionRequested; //
WSADATA  wsaData;      //
int err;
wVersionRequested = MAKEWORD(1,1);
err = WSAStartup(wVersionRequested,&wsaData);
if( err != 0)
{
return;
}
if(LOBYTE( wsaData.wVersion ) != 1||
HIBYTE( wsaData.wVersion) != 1)
{
WSACleanup();
return;
}
SOCKET sockSrv=socket(AF_INET,SOCK_STREAM,0); //
SOCKADDR_IN addrSrv;
addrSrv.sin_addr.S_un.S_addr=htonl(INADDR_ANY);
addrSrv.sin_family=AF_INET;
addrSrv.sin_port=htons(6000);
bind(sockSrv,(SOCKADDR*)&addrSrv,sizeof(SOCKADDR));  //
listen(sockSrv,5);   //     socket
SOCKADDR_IN addrClient;
int len=sizeof(SOCKADDR);
while(1)  //
{
SOCKET sockConn=accept(sockSrv,(SOCKADDR*)&addrClient,&len);
char sendBuf[100];
sprint(sendBuf,"Welcome %s to http://www.sunxin.org/",
inet_ntoa(addrClient.sin_addr));
send(sockConn,sendBuf,strlen(sendBuf)+1,0);
char recvBuf[100];
recv(sockConn,recvBuf);
printf("%sn",recvBuf);
closesocket(sockConn);
WSACleanup();
}
}
```

Server File->New->Win32 Console Application TcpSrv File->New->C++ Source File TcpSrv Setting Link Object/library modules ws2_32.lib

```
#include <Winsock2.h>
#include <stdio.h>
void main()
{
WORD  wVersionRequested;
WSADATA  wsaData;
int err;
wVersionRequested = MAKEWORD(1,1);
err = WSAStartup(wVersionRequested,&wsaData); //    winsock Dll
if( err != 0)
{
return;
}
if(LOBYTE( wsaData.wVersion ) != 1||
HIBYTE( wsaData.wVersion) != 1)
{
WSACleanup();
return;
}
SOCKET sockClient=socket(AF_INET,SOCK_STREAM,0);
SOCKADDR_IN addrSrv;
addrSrv.sin_addr.S_un.S_addr=inet_addr("127.0.0.1");
addrSrv.sin_family=AF_INET;
addrSrv.sin_port = htons(6000);
connect(sockClient,(SOCKADDR*)&addrSrv,sizeof(SOCKADDR));
char recvBuf[100];
recv(sockClient,recvBuf,100,0);
printf("%sn",recvBuf);
send(sockClient,"This is zhangsan",strlen("This is zhangsan")+1,0);
closesocket(sockClient);
WSACleanup();
}
```

Client File->New->Win32 Console Application TcpClient File->New->C++ Source File TcpClient Setting Link Object/library modules ws2_32.lib

8. ( )..

C

```
#include <iostream.h>
class human
{
public:
   human(){ human_num++;}; //
   static int human_num;     //
   ~human()
   {
     human_num--;
     print();
   }
   void print()           //
   {
    cout<<"human num is: "<<human_num<<endl;
   }
 protected:
 private:
};
int human::human_num = 0;   //                   ,
human f1(human x)
{
   x.print();
   return x;
}
int main(int argc, char* argv[])
{
 human h1; //                ,human_num     1
 h1.print(); //      Human_man:1
 human h2 = f1(h1); //        f1(),    human_num:1,        human_num    0,
 h2.print();//         :human_num:0
 return 0;
} //                       :     :human_num:-1,human_num:-2;
```

调用f1()后, return时会调用析构函数,故会变为0

```
1
1
0
0
  1
  2
```

```
human h1; //                ,---hum_num = 1;
h1.print(); //     :"human is 1"
human h2 = f1(h1); //        f1(h1)          ,                              ,
  ,h2                              ,
```

C/C++

```
}
```

1 str1 11 ’’ string 10 strcpy

2 str1 3 strcpy(string, str1) str1 string 7 strcpy 10

3 if(strlen(str1) <= 10) if(strlen(str1) < 10) strlen ’’ 1

(1) ’\0’
(2)
(3) strcpy strcpy 10

2

```
void strcpy( char *strDest, char *strSrc )
{
   while( (*strDest++ = * strSrc++) != '' );
}
```

4

```
void strcpy( char *strDest, const char *strSrc )
//              const                        2
{
   while( (*strDest++ = * strSrc++) != '' );
}
```

7

```
void strcpy(char *strDest, const char *strSrc)
{
  //                          0          3
  assert( (strDest != NULL) && (strSrc != NULL) );
  while( (*strDest++ = * strSrc++) != '' );
}
```

10

```
//                                    3
```

```
  while( (*strDest++ = * strSrc++) != '' );
    return address;
}
```

2 10 strcpy strcpy

(4) strlen ''

strcpy 10 strlen int strlen(const char *str) // const

```
{
  assert( strt != NULL ); //                    0
  int len;
  while( (*str++) != '' )
  {
    len++;
  }
  return len;
}
```

4

```
void GetMemory( char *p )
{
  p = (char *) malloc( 100 );
}

void Test( void )
{
  char *str = NULL;
  GetMemory( str );
  strcpy( str, "hello world" );
  printf( str );
}
```

5

```
char *GetMemory( void )
{
  char p[] = "hello world";
  return p;
}

void Test( void )
{
  char *str = NULL;
  str = GetMemory();
  printf( str );
}
```

6

```
void GetMemory( char **p, int num )
{
  *p = (char *) malloc( num );
}

void Test( void )
{
  char *str = NULL;
  GetMemory( &str, 100 );
  strcpy( str, "hello" );
  printf( str );
}
```

7

```
void Test( void )
{
  char *str = (char *) malloc( 100 );
  strcpy( str, "hello" );
  free( str );
  ... / /
}
```

4 GetMemory(char *p)

```
char *str = NULL;
GetMemory( str );
```

str NULL

5

```
char p[] = "hello world";
return p;
```

p[]

6 GetMemory 4 GetMemory

GetMemory

```
*p = (char *) malloc( num );
```

```
if ( *p == NULL )
```

```
{
  ...//
}
```

7 6

```
char *str = (char *) malloc(100);
```

free(str) str “ ”

```
str = NULL;
```

6 Test malloc

4 7 50~60

1

2

3

```
swap( int* p1,int* p2 )
{
  int *p;
  *p = *p1;
  *p1 = *p2;
  *p2 = *p;
}
```

swap p “ ” VC++ DEBUG
“Access Violation”

```
swap( int* p1,int* p2 )
{
  int p;
  p = *p1;
  *p1 = *p2;
  *p2 = p;
}
```

3.

1 BOOL int float “ ” if var

BOOL if(!var)
int if(var==0)
float
const float EPSINON = 0.00001;

if ((x >= - EPSINON) && (x <= EPSINON)

if(var==NULL)

0 “ ” BOOL 0 if(var==0) int
if(!var) if(!var)

if “ ” “ ” if(var) if(!var) “ ”
if (short int long) if(var==0) 0 “ ”
if(var==NULL)

float “==” “ =” “>=” “<=”
if (x == 0.0) 0

2 Windows NT 32 C++ sizeof

```
void Func ( char str[100] )
{
  sizeof( str ) = ?
}

void *p = malloc( 100 );
sizeof ( p ) = ?
```

```
sizeof( str ) = 4
sizeof ( p ) = 4
```

Func (char str[100])

1

```
char str[10];
cout << sizeof(str) << endl;
```

10 str char[10]

2

```
char str[10];
str++; //          str
```

3

Windows NT 32 4 sizeof(str) sizeof (p)
4

3 “ ” MIN

```
least = MIN(*p++, b);
```

```
#define MIN(A,B) ((A) <= (B) ? (A) : (B))
```

MIN(*p++, b)

“ ” “ ”

1 “ ”

```
#define MIN(A,B) (A) <= (B) ? (A) : (B)
#define MIN(A,B) (A <= B ? A : B )
```

0

2

#define MIN(A,B) ((A) <= (B) ? (A) : (B)) MIN(*p++, b)
((*p++) <= (b) ? (*p++) : (*p++))
p ++
0

```
#define MIN(A,B) ((A) <= (B) ? (A) : (B));
```

“;” 0

4

```
#ifndef __INCvxWorksh
#define __INCvxWorksh
#ifdef __cplusplus

extern "C" {
#endif
/*...*/
#ifdef __cplusplus
}

#endif
#endif /* __INCvxWorksh */
```

```
#ifndef  __INCvxWorksh
#define  __INCvxWorksh
#endif
```

C++ C C++ symbol C

```
void foo(int x, int y);
```

C symbol _foo C++ _foo_int_int _foo_int_int C++

C C++ C++ C extern "C" extern "C" C _foo C C++

5 char n “abcdefghi” n=2 “hiabcdefgh”

```
//pStr       ''
//steps        n

void LoopMove ( char * pStr, int steps )
{
  //    ...
}
```

1

```
void LoopMove ( char *pStr, int steps )
{
  int n = strlen( pStr ) - steps;
  char tmp[MAX_LEN];
  strcpy ( tmp, pStr + n );
  strcpy ( tmp + steps, pStr);
  *( tmp + strlen ( pStr ) ) = '';
  strcpy( pStr, tmp );
}
```

2

```
void LoopMove ( char *pStr, int steps )
{
  int n = strlen( pStr ) - steps;
  char tmp[MAX_LEN];
  memcpy( tmp, pStr + n, steps );
  memcpy(pStr + steps, pStr, n );
  memcpy(pStr, tmp, steps );
}
```

1 strcpy

2 memcpy

3 memset

6 WAV WAV WAV WAV

WAVE

```
  char cRiffFlag[4];
  UIN32 nFileLen;
  char cWaveFlag[4];
  char cFmtFlag[4];
  char cTransition[4];
  UIN16 nFormatTag ;
  UIN16 nChannels;
  UIN16 nSamplesPerSec;
  UIN32 nAvgBytesperSec;
  UIN16 nBlockAlign;
  UIN16 nBitNumPerSample;
  char cDataFlag[4];
  UIN16 nAudioLength;

} WAVEFORMAT;
```

WAV buffer

```
WAVEFORMAT waveFormat;
memcpy( &waveFormat, buffer,sizeof( WAVEFORMAT ) );
```

waveFormat WAV

6

memcpy memset

7 String String

```
class String
{
  public:
    String(const char *str = NULL); //
    String(const String &other); //
    ~ String(void); //
    String & operate =(const String &other); //
  private:
    char *m_data; //
};
```

```
//

String::String(const char *str)
{
  if(str==NULL)
  {
    m_data = new char[1]; //                                      ''
    //            m_data    NULL
    *m_data = '\0';
  }
  else
  {
    int length = strlen(str);
    m_data = new char[length+1]; //          NULL
    strcpy(m_data, str);
  }
}

// String

String::~String(void)
{
  delete [] m_data; //     delete m_data;
}

//

String::String(const String &other)         //                       const
{
  int length = strlen(other.m_data);
  m_data = new char[length+1];           //            m_data    NULL
  strcpy(m_data, other.m_data);
}

//

String & String::operate =(const String &other) //                      const
{
  if(this == &other)      //
    return *this;
  delete [] m_data;            //
  int length = strlen( other.m_data );
  m_data = new char[length+1];    //            m_data    NULL
  strcpy( m_data, other.m_data );
  return *this;                    //
}
```

String

C++ 60%

m_data

C++ Effective C++

60% C++

8 static const

static n

1 static auto

2 static

3 static

4 static

5 static this static

const n

1 const const

2 const const
const

3 const

4 const

5 const “ ”

```
const classA operator*(const classA& a1,const classA& a2);
```

operator* const

```
classA a, b, c;
(a * b) = c; //      a*b
```

(a * b) = c

static const 1~2

static const

4.

1 C Big_endian 0 Little_endian 1

```
int checkCPU()
{
  {
    union w
    {
      int a;
      char b;
    } c;
    c.a = 1;
    return (c.b == 1);
  }
}
```

Little-endian Big-endian Little-endian CPU
Big-endian

16bit 0x1234 Little- endian CPU 0x4000

0x4000	0x34
0x4001	0x12

Big-endian CPU

0x4000	0x12
0x4001	0x34

32bit 0x12345678 Little-endian CPU 0x4000

0x4000	0x78
0x4001	0x56
0x4002	0x34
0x4003	0x12

Big-endian CPU

0x4000	0x12
0x4001	0x34
0x4002	0x56
0x4003	0x78

union CPU
Little-endian Big-endian

2 1+2+3+…+n

```
int Sum( int n )
{
  return ( (long)1 + n) * n / 2;      //    return (1l + n) * n / 2;
}
```

“ ” return (1 l + n) * n / 2

```
int Sum( int n )
{
 long sum = 0;
 for( int i=1; i<=n; i++ )
 {
   sum += i;
 }
 return sum;
}
```

c/c++ java .

c/c++

.

4 SQL

SELECT * FROM TABLE SELECT * FROM TABLE
WHERE NAME LIKE '%%' AND ADDR LIKE '%%'
AND (1_ADDR LIKE '%%' OR 2_ADDR LIKE '%%'
OR 3_ADDR LIKE '%%' OR 4_ADDR LIKE '%%')

: NAME null ADDR=null 1_ADDR LIKE 2_ADDR
3_ADDR 4_ADDR null.
NAME ADDR Null

5 SQL

1 g_cardapply
(/ /)
g_applyno varchar 8 //
g_applydate bigint 8 //
g_state varchar 2 //
2 g_cardapplydetail
(/ /)
g_applyno varchar 8 //
g_name varchar 30 //
g_idcard varchar 18 //
g_state varchar 2 //

1 440401430103082

```
select A.g_applydate
from g_cardapply A inner join g_cardapplydetail B on A.g_applyno = B.g_applyno
where B.g_idCard = '440401430103082'
```

2

```
select g_idCard,count(*) as Cnt from g_cardapplydetail
group by g_idcard
having count(*) > 1
```

3 440401430103082 07

```
update g_cardapplydetail set g_state = '07'
where g_idcard = '440401430103082'

update A set g_state = '07'
from g_cardapply A inner join g_cardapplydetail B on A.g_applyno = B.g_applyno
where B.g_idcard = '440401430103082'
```

4 g_cardapplydetail

```
delete from g_cardapplydetail
where g_name like '   %'
```

3 440401430103082 07

```
update g_cardapplydetail set g_state = '07'
where g_idcard = '440401430103082'

update A set g_state = '07'
from g_cardapply A inner join g_cardapplydetail B on A.g_applyno = B.g_applyno
where B.g_idcard = '440401430103082'
```

5 SQL :

```
/*Select g_cardapply. g_applydate
From g_cardapply, g_cardapplydetail
Where g_cardapply. g_applyno=g_cardapplydetail. g_applyno
And g_cardapplydetail.g_idcard='440401430103082'*/

/*Select *From (select count(*) g_count , g_idcard
From g_cardapplydetail
Group by g_idcard ) a
Where a. g_count >= 2*/

/*Update g_cardapply
set g_state='07'
where g_applyno in (select distinct g_applyno
from g_cardapplydetail
where g_idcard ='440401430103082')
update g_cardapplydetail
set g_state='07'
where g_idcard='440401430103082' */

/*Delete from g_cardapplydetail
Where g_name like '   %'*/
```

PS: GF

4. In C++, there're four type of Casting Operators, please enumerate and explain them especially the difference.

C++

reinterpret_cast,static_cast,const_cast,dynamic_cast

static_cast

dynamic_cast

const_cast const

reinterpret_cast orimplmentation_dependent

7

```
#include <iostream>
#include <vector>
using namespace std;
void print(vector<int>);
int main()
{
vector<int> array;
array.push_back(1);
array.push_back(6);
array.push_back(6);
array.push_back(3);
//      array              6
vector<int>::iterator itor;
vector<int>::iterator itor2;
itor=array.begin();

for(itor=array.begin(); itor!=array.end(); )
{
if(6==*itor)
{
itor2=itor;
array.erase(itor2);
}
itor++;
}
print(array);
return 0;
}
void print(vector<int> v)
{
cout << "n vector size is: " << v.size() << endl;
vector<int>::iterator p = v.begin();
}
```

6

```
void print(const vector<int>&);
int main()
{
```

```
vector<int> array;
array.push_back(1);
array.push_back(6);
array.push_back(6);
array.push_back(3);

//      array                 6
array.erase( remove( array.begin(), array.end(), 6 ) , array.end() );

print(array);
return 0;
}
void print(const vector<int>& v)
{
cout << "n vector size is: " << v.size() << endl;
copy(v.begin(), v.end(), ostream_iterator<int>(cout, " ") );
}
#include <iostream>
#include <vector>
using namespace std;
int main()
{
vector<int> array;
array.push_back(1);
array.push_back(6);
array.push_back(6);
array.push_back(6);
array.push_back(6);
array.push_back(6);
array.push_back(3);
array.push_back(9);
array.push_back(8);
array.push_back(5);
//&Eacute;&frac34;&sup3;&yacute;array&Ecirc;&yacute;×é&Ouml;&ETH;&Euml;ù&Oacute;&ETH;&micro;&Auml;6
vector<int>::iterator itor;
itor=array.begin();
for(itor=array.begin(); itor!=array.end();++itor )
{
if(6==*itor)
{
itor=array.erase(itor);
--itor;
}
}
cout << "vector size is: " << array.size() << endl;
for(itor=array.begin(); itor!=array.end();++itor )
{
```

```
cout<<*itor<<" ";
}
system("pause");
return 0;
}
```

: itor=array.erase(itor); itor

6 2 6 itor++ array.end()

1 6 6 3 array.end() // itor 6

1 6 3 array.end() // 6 6

1 6 3 array.end() //itor++ 3

2. What are three ways in which a thread can enter the waiting state?

:CPU thread thread waiting

thread waiting

2

1

2

hr , hr

hr n

n>=1 ----

1 hr

m m>=2 Hr 2 1

game over

1 ok,

1.

2.

3.

4. strstr()

5.

6.

7.

8. 4 ,

9.

10. malloc()

11. A B A B

12.

13.

14.

12 2 24

1. C++ D

A. int B. char

C. bool D. real

2. break C

A. for B. switch

C. if D. while

3. B

A. B.

C. D.

4. extern D

A. B.

C. C++ D.

5. C strstr A

A. B.

C. D.

6. stl::deque A

A. B.

C. D.

7. STL D

A. KMP B.

C. D.

8. A

A. B.

C. D.

9. Win32 C

A. THREAD_PRIORITY_HIGHEST

B. THREAD_PRIORITY_IDLE ,

C. THREAD_PRIORITY_TIME_CRITICAL

D. THREAD_PRIORITY_ABOVE_NORMAL

10. WinMain D

A. HINSTANCE B. INT

C. LPSTR D. WPARAM

11. VC++ new B

A. VirtualAlloc() B. HeapAlloc()

C. GlobalAlloc() D. AllocateUserPhysicalPages()

12. C++ C

A. Think in C++ B. MFC

C. C++ Primer D. Effective C++

13. Windows API InvalidateRect, A

A. WM_PAINT B. WM_CREATE

C. WM_NCHITTEST D. WM_SETFOCUS

14. virtual void Draw()=0 C

(1) ( )

(2) ( )

(3) ( )

(4) ADT ( )// ,

.

A. 1 B. 2

C. 3 D. 4

6 3 18

1. vector::iterator ACD

A. ++ B. >>

C. * D. ==

2. CreateFile( ) AB

A. B.

C. D.

3. HANDLE ABCD

A. HINSTANCE B. HWND

C. HDC D. HFONT

4. OpenGL A

A. GL_FOG B. GL_LINE_STRIP

C. GL_POINTS D. GL_TRIANGLE_FAN

5. ABD

A. “::” B. “.”

C. “->” D. “? :”

6. C++

A. Bill Gates B. Stanley Lippman

C. Anderson Hejlsberg D. Bjarne Stroustrup

7. ABC

A. ifndef/define/endif

B. #include <filename.h> filename.h

#include “filename.h” filename.h

C. C++ C C++ C

void foo(int x, int y); C _foo C++

_foo_int_int C++ C extern“C”

D. fopen // ,fopen

8 3 24

1. 320 X 192 ______ 24

______

2. Windows API ____ windows ___

3. __asin()____ __fabs()____ n

__pow()____ ___fcvt()___

4. i 5 (++i) - - __6__

5. API LoadBitmap() __ __

6. new __delete___ malloc _free___ _ __ calloc

__ ___ realloc _ _

7. SendMessage PostMessage SendMessage__

, __ PostMessage__ , ___

8.

```
const int nCount = 12;
const double dOrgX = 5.0,
dOrgY = 3.0;
const double dRadius = 2.0;
for( int i = 0; i < nCount; i++ )
{
double dAngle = M_PI * 2.0 / (double)nCount * i;
cout << "  " << i << "    X = " << ________; cout << ", Y = " << __________ << endl;
}
```

12 2 24

1.

2.

3.

4. const

5.

6. char *p = "Test"; p[0] = 'R';

7. cout << "Test";

8. stl::list

9. stl::vector stl::list

10. VC VC++ VC++ C++

11. new malloc

12. C++ struct class

(3 5 15)

1. PeekMessage GetMessage

: Peekmessage Getmessage

, Getmessage OS ,

; Peekmessage , OS

, ,GetMessag() , PeekMessage()

wRemoveMsg

2. Windows SDK

: SetTimer ,SetTimer

UINT_PTR SetTimer(HWND hWnd, UINT_PTR nIDEvent, UINT uElapse, TIMERPROC lpTimerFunc

3. const

: const

const

const

const

const

3 1 7 2 14 3 24

1.

```
struct Node
{
Node *Parent;

Node *Left, *Right;

};
void Through(Node *Root)
{
}
```

2.

```
int DicFind( int *Array, int Count, int Value )
{

}
```

3. String

String

```
class String
{
public:
String( const char *pStr = NULL ); //
~String( void ); //
String &operate = ( const String &Source ); //
private:
char *m_pData; //
};
```

1 3 3 4 0 9

2 1 2 3 4 5

```
public static int getAge() {
int age;
int third;
int fourth;

for (int i = 11; true; i++) {
if (i < 200) {
third = (int) Math.pow(i, 3);
fourth = (int) Math.pow(i, 4);
if (getLength(third, fourth) == 10) {
age = i;
break;
}
```

```
}
}
return age;
}

public static int getLength(int args1, int args2) {
String str1 = String.valueOf(args1);
String str2 = String.valueOf(args2);
String str = str1 + str2;
if (str.length() != 10) {
return -1;
}
int[] intarray = new int[10];
for (int i = 0; i < str.length(); i++) {
intarray[i] = Integer.parseInt(str.substring(i,i+1));
}
Arrays.sort(intarray);
if(intarray[0]!=0 && intarray[9]!=9)
return -1;

return 10;
}


for(int i=1; i<6; i++){
for(int j=1; j<6; j++){
if(i==j){
System.out.println(j+""+j);
}else{
System.out.println(i+""+j);
System.out.println(j+""+i);
}
}
}
public class A {
// http://community.csdn.net/Expert/topic/4667/4667929.xml?temp=.57922
public static void main(String[] args) {
String t;
String[] s = new String[5];
int j = s.length;
for(int i=0; i<j; i++) {
s[i] = new Integer(i+1).toString();
}

for(int i=0; i<j; i++) {
t = s[i];
for(int a=0; a<j; a++) {
t += s[i];
System.out.println(t);
}
System.out.println();
}
}
}
```

```
for(int i=1; i<6; i++){
for(int j=1; j<6; j++){
if(i==j){
System.out.println(j+""+j);
}else{
System.out.println(i+""+j);
System.out.println(j+""+i);
}
}
}
```

==
===

1 2 3 4 5

```
for(int i=1; i<6; i++){
for(int j=1; j<6; j++){
if(i==j){
break;
}else{
System.out.println(i+""+j);
System.out.println(j+""+i);
}
}
}
public class B {
public static void main(String[] args) {
for (int i = 1; i < 6; i++) {
int t = i;
for(int a = 0; a<5; a++) {
int c = a+1;
if(c == t) {
continue;
}else {
System.out.println(t*10+c);
}
}
System.out.println();
}
}
```

```
}

public class Test
{
public static void main(String[] args)
{
int[][] a=new int[5][];
for(int i=0;i<a.length;i++)
{
a[i]=new int[i+1];
}
for(int i=1;i<=a.length;i++)
{

for(int j=i+1;j<=a.length;j++)
{
System.out.print(i);
System.out.print(j+" ");
}
System.out.print(" ");
}

for(int i=a.length;i>0;i--)
{

for(int j=i-1;j>0;j--)
{
System.out.print(i);
System.out.print(j+" ");
}
System.out.print(" ");
}
}
}


public class Test {


public static int getDigits(String str) {

int[] intarr = new int[10];

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

intarr[i] = 0;

for (int i = 0; i < str.length(); i++) {

int j = Integer.parseInt(str.substring(i, i + 1));

intarr[j] = 1;
}
int num = 0;
for (int i = 0; i < 10; i++)
num = num + intarr[i];
return num;
}

private static int getAge() {
int age;
int third;
int fourth;
for (age = 1; age < 100; age++) {
third = (int) Math.pow(age, 3);
fourth = (int) Math.pow(age, 4);
if (third < 1000 || third >= 10000)
continue;
if (fourth < 100000 || fourth >= 1000000)
continue;
String str = String.valueOf(third) + String.valueOf(fourth);
if (getDigits(str) == 10)
return age;
}
return 0;
}
}
```

```
class Combine
{
public static void main(String[] args)
{
for(int i=1; i<5; i++)
{
for(int j=i+1; j<6; j++)
{
System.out.println(i+""+j);
System.out.println(j+""+i);
}
}
}
public class Age
{
public static void main(String [] args)
{
String str1 = null;
String str2 = null;
String str3 = null;
String str4 = "0123456789";
for(int i=10;i<50;i++)
{
str1 = Integer.toString(i*i*i);
str2 = Integer.toString(i*i*i*i);
str3 = str1+str2;
if((str1.length() == 4) && (str2.length() ==6))
{
boolean flag = true;
for(int j=0;j<10;j++)
if(str3.indexOf(str4.charAt(j))==-1)
flag = false;
if(flag){
System.out.println(">>>"+i);
System.out.println(str3);
}
}
}

}
}
```

~

silver6 | 04 , 2006 09:48

4

4 *_#!~

```
long atol(char *str)
{
char c = *str;
if( !isdigit(c) ) str++;
for(long value = 0; *str != ''; value = value * 10 + (*str -'0'),str++);
return c == '-' ? -value : value ;
}
void stol(const char * des, long& num)
{
for (int base = 1, i = 0; des[i] != ''; base = 10, ++i)
{
num *= base;
num += (int)(des[i] - '0');
}
}
```

num   0

```
void stol(const char * des, long& num)
{
for (int i=num=0; des[i] != ''; i++)
{
num *= 10;
num += (int)(des[i] - '0');
}
}
void stol(char *str, long &num)
{
while(*str != '')
{
num = num * 10 + (*str - '0');
str++;
}
}
void stol(const char * des, long& num)
{
char p = des[0];
for (int b = 1, pos = 1, base = 1; des[pos] != ''; b = 10, ++pos, base *= 10)
{
(num *= b) += (int)(des[pos] - '0');
}
p == '-' ? (num *= -1) : (num = (int)(des[0] - '0') * base + num);
}
```

```
#include <iostream>
using namespace std;
long str2long(char* p,long xxx=0L)
{
return *p==''?xxx:str2long(p,xxx*10+(*p+++0-'0'));
}

int main()
{
char *str="123456789",*p=str;
cout<<str2long(p);
getchar();
return 0;
}
```

STL

```
#include <sstream>
#include <iostream>
#include <string>
using namespace std;
long ToLong(string& s)
{
long l;
istringstream iss(s);
iss>>l;
return l;
}
int main(int argc, _TCHAR* argv[])
{
string s = "-12356";
cout<<ToLong(s);
return 0;
}

#include <iostream>
using namespace std;

long str2long(char* p,long xxx=0L,bool IsPositive=true)
{
return
*p==''?(IsPositive?xxx:xxx*(-1)):(*p=='-'?str2long(++p,0L,false):str2long(p,xxx
*10+*p+++0-'0',IsPositive));
}

int main()
{
char *str="-123456789",*p=str;
cout<<str2long(p);
getchar();
return 0;
}
```