指针：

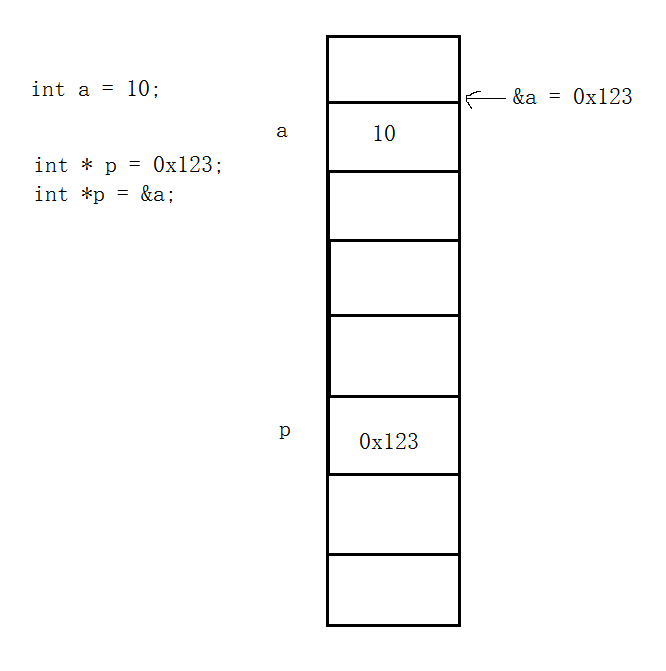
1、定义：

① 地址：人为对于每个字节进行编号，该编号即为地址

② 指针：就是地址

③ 指针变量：保存地址的变量

一般所说的指针即为上述三者中任意一个，不做区分



2、格式

存储类型 数据类型 \* 指针变量名;

注：数据类型 表示指向空间的内容的数据类型

数据类型 \* 表示指向空间地址的数据类型

指针变量名 表示指向空间的地址

\* 指针变量名 表示指向空间的内容

3、指针使用

指针变量名 表示指向空间的地址

\* 指针变量名 表示指向空间的内容

4、初始化

指针必须初始化，否则指针随意指向会出现”野指针”，”野指针”存在安全隐患，应避免出现

指针可以使用NULL进行初始化，NULL表示特殊地址，该地址其中保存特殊值 0

int \*p = NULL;

5、指针的修饰

变量必须使用同类指针操作

（1）万能指针

void \*p;

该指针可以接收任意类型地址，但使用时需要进行强制类型转换

（2）const **（笔试原题）**

const常量化：const修饰的变量具有常量特征，即只能读不能写

const int \*p;（指针常量） 不能通过指针修改指向空间的内容，但是可以修改指针的指向

int \* const p;（常量指针） 不能修改指针的指向，但是可以通过指针修改指向空间内容

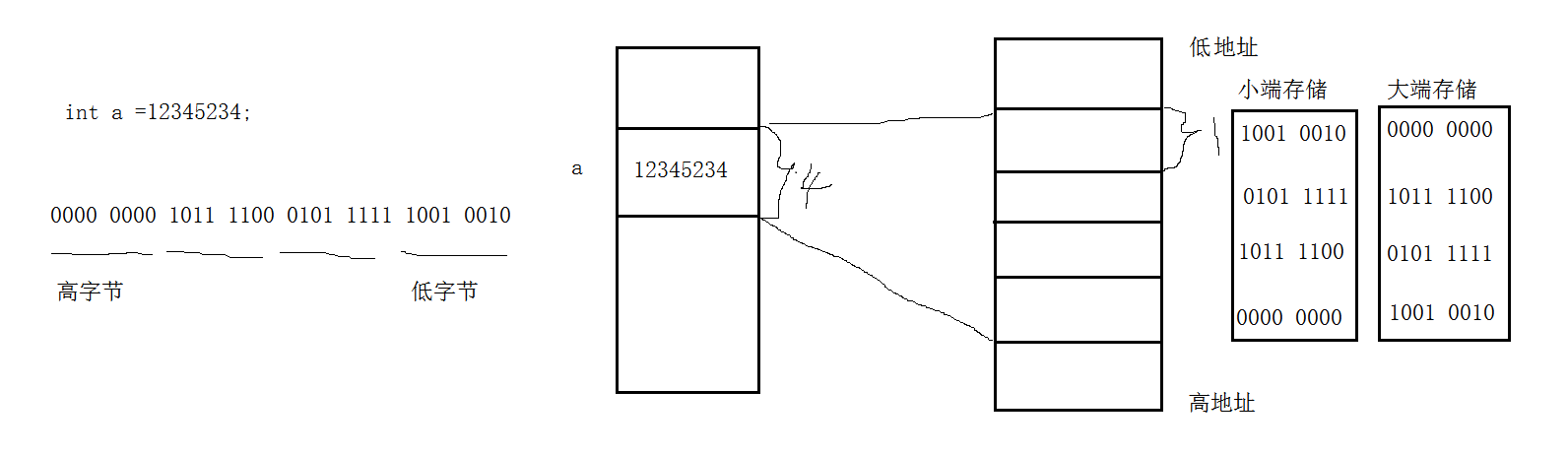
const int \* const p; 指针指向和指针指向空间内容均不可修改

6、指针大小**（笔试原题）**

在32位操作系统下，所有类型指针其大小均为4字节；在64位操作系统下，所有类型指针其大小均为8字节

7、大小端存储

数据在内存上存储时，若低字节存放在低地址，高字节存放在高地址，则称为小端存储;若低字节存放在高地址，高字节存放在低地址，则称为大端存储



**笔试原题：**验证大小端存储

8、字符指针

字符指针既可以指向字符变量，也可以指向字符串常量

字符指针保存字符串首地址，即字符指针直接指向常量区，因此不能使用指针修改字符串

9、指针运算

前提：① 必须在同一段连续内存上操作

② 必须是同一类指针（同类型指针操作内存范围相同）

③ \*地址 取该地址空间的值 &变量 取该变量空间地址

④ a[i] == \*(a + i) ==>> [] == \*( + )

指针操作数组：

int a[10] = {10,20,30,40,50,60,70,80,90,100};

元素地址：&a[i] int \*

a + i int \*

int \*p1 = a + 6,\*p2 = a + 3;

（1）算术运算

p1 - p2：表示两个指针相差|p1 - p2|个单位，值为正数，表示p1地址大，p2地址小，值为负数则相反

p2 + m：表示p2向地址大端偏移m个单位

p2 - m：表示p2向地址小端偏移m个单位

p2++/++p2：表示p2向地址大端偏移1个单位

p2--/--p2：表示p2向地址小端偏移1个单位

（2）赋值运算符

p1 = p2：表示p1和p2指向同一块内存

p1 +=m：表示p1向地址大端偏移m个单位

（3）关系运算符

p1 > p2：表示p1地址大，p2地址小

p1 < p2：表示p2地址大，p1地址小

p1 == p2：表示p1和p2指向同一块内存

作业：

1、有 N个国家名，要求按字母先后顺序排列（用起泡排序法）后输出。

#include<stdio.h>

#include <string.h>

#define N 5

int main()

{

char a[N][100]={"endland","china","australia","canada","germany"};

int i,j;

//冒泡排序法

for(i=0;i<N-1;i++)

{

for(j=0;j<N-1-i;j++)

{

if(strcmp(a[j],a[j+1])>0) //交换

{

char b[100];

strcpy(b,a[j]);

strcpy(a[j],a[j+1]);

strcpy(a[j+1],b);

}

}

}

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

{

printf("%s\n",a[i]);

}

return 0;

}

1. 把一个字符串的大写字母放到字符串的后面，且不能申请额外的空间。

#include<stdio.h>

#include<string.h>

int main()

{

char a[100]="saerAREzsgtRfsg",t;

int i,j;

int len=strlen(a);

for(i=0;i<len;i++)//先确定最后一位

{

for(j=0;j<len-1-i;j++)//okk

{

if(a[j]>='A'&&a[j]<='Z' &&a[j+1]>='a'&&a[j+1]<='z')

{

t=a[j];a[j]=a[j+1];a[j+1]=t;//出现Ab-> 交换

}

}

}

printf("%s\n",a) ;

return 0;

}

#include<stdio.h>

#include<string.h>

int main()

{

char a[100]="adfsfAEGkcgjhBY";

char b[100],c[100];

int i=0,j=0,k=0;

while(a[i]!='\0')

{

if(a[i]>='A'&& a[i]<='Z')

{

b[j]=a[i];

j++;

}

if(a[i]>='a'&& a[i]<='z')

{

c[k]=a[i];

k++;

}

i++;

}

b[j]='\0';c[k]='\0';

strcat(c,b);

printf("%s\n",c);

return 0;

}

一、选择题

1、 若有以下定义和语句，且0≤i<10则对数组元素的错误引用是（ C ）

int a[10]={1,2,3,4,5,6,7,8,9,10},\*p,i;

p=a;

A. \*(a+i) B. a[p-a] C. p+i D. \*(&a[i])

C \*p+i

4、

char \*s="\t\\Name\\Address\n"; D

指针s所指字符串的长度为： 。

A. 说明不合法 B. 19 C. 18 D. 15

8、 若int x ，\*pb； ，则正确的赋值表达式是 A 。

A. pb=&x B. pb=x； C. \*pb=&x； D. \*pb=\*x

9、 有如下程序段

int \*p ，a=10 ，b=1 ；

p=&a ； a=\*p+b ；

执行该程序段后，a 的值为 B 。

A. 12 B. 11 C. 10 D. 编译出错

10、若有以下定义和语句：

double r=99 ， \*p=&r ；

\*p=r ；

则以下正确的叙述是 B 。

A. 以下两处的\*p 含义相同，都说明给指针变量p 赋值

B. 在"double r=99,\*p=&r；"中，把r 的地址赋值给了p 所指的存储单元

C. 语句"\*p=r；"把变量r 的值赋给指针变量p

D. 语句"\*p=r；"取变量r 的值放回r 中

12、若有说明：long \*p ，a；则不能通过scanf 语句正确给输入项读入数据的程序段

是 A 。

A. \*p=&a； scanf（"%ld"，p）;

B. p=（long \*）malloc（8）; scanf（"%ld"，p）;

C. scanf（"%ld"，p=&a）;

D. scanf（"%ld"，&a）;

\*p=&a; scanf("%ld",p); //应修改为p=&a; scanf("%ld",p);  
B, p=(long \*)malloc(8); scanf("%ld",p);//给p分配空间，然后给它指定值  
C, scanf("%ld",p=&a);//先做p = &a,然后再对\*p赋值  
D, scanf("%ld",&a);//跟p无关，纯属对a的操作

13、对于类型相同的两个指针变量之间，不能进行的运算是 C 。

A. < B. = C. + D. –

类型相同的两个指针变量之间可以进行<（小于）、=（等于）、-（减法）运算。

14、若已定义：int a[9] ，\*p=a；并在以后的语句中未改变p 的值，不能表示a[1] 地

址的表达式是 C 。

A. p+1 B. a+1 C. a++ D. ++p

A、p+1 //p指向a[0]，p+1显然就是a[1]的地址。

B、a+1 //a指向a[0]，a+1显然就是a[1]的地址。

C、a++  //这个先取a的值，再使a自加1，而a的值是a[0]的地址，因此本题答案就是这个。

D、++p  //这个表示先使p自加，再取值。p指向a[0]，自加后指向a[1]，因此++p也是a[1]的地址。

15、若有以下说明：

int a[10]={1，2，3，4，5，6，7，8，9，10} ，\*p=a ；

则数值为6 的表达式是 C 。

A. \*p+6 B. \*（p+6） C. \*p+=5 D.p+5 ->a[5]的地址

16、设P1 和P2 是指向同一个int 型一维数组的指针变量，k 为int 型变量，则不能正

确执行的语句是 B 。

A. k=\*P1+\*P2; B. p2=k; C. P1=P2; D. k=\*P1 \* （\*P2）;

17、若有以下的定义：

int a[ ]={1 ，2 ，3 ，4 ，5 ，6 ，7 ，8 ，9 ，10} ， \*p=a ；

则值为3 的表式是 A 。

A. p+=2 ， \*（p++） B. p+=2 ，\*++p

C. p+=3 ， \*p++ D. p+=2 ，++\*p

18、若有以下定义和语句：

int a[10]={1 ，2 ，3 ，4 ，5 ，6 ，7 ，8 ，9 ，10} ，\*p=a ；

则不能表示a 数组元素的表达式是 B 。

A. \*p B. a[10] C. \*a D. a[p-a]

19、有如下说明

int a[10]={1，2，3，4，5，6，7，8，9，10}，\*p=a；

则数值为9 的表达式是 B 。

A. \*p+9 B. \*（p+8） C. \*p+=9 D. p+8

20、下面程序输出数组中的最大值，由s 指针指向该元素.

void main（）

{ int a[10]={6，7，2，9，1，10，5，8，4，3，}，\*p，\*s；

for（p=a， s=a； p-a<10； p++）

if（【 】）s=p；

printf（"The max：%d"，\*s）：

}

则在if 语句中的判断表达式应该是 B 。

A. p>s B. \*p>\*s C. a[p]>a[s] D. p-a>p-s

21、若有以下定义和语句：

int a[10]={1 ，2 ，3 ，4 ，5 ，6 ，7 ，8 ，9 ，10} ，\*p=a ；

则不能表示a 数组元素的表达式是 B 。

1. \*p B. a[10] C. \*a D. a[p-a]

24、设有如下的程序段：

char str[ ]="Hello" ；

char \*ptr ；

ptr=str ；

执行上面的程序段后 ，\*（ptr+5）的值为 B 。

A. 'o' B. '\0' C. 不确定的值 D. 'o'的地址

26、下面各语句行中，不能正确进行字符串赋值操作的语句是 A 。

A. char ST[5]={"ABCDE"}； B. char S[5]={'A'，'B'，'C'，'D'，'E'}；

C. char \*S； S="ABCDE"； D. char \*S； scanf（"%S"，S）；

char s[5]="abcde"的时候后面的被认为是字符串，所以会以‘\0’结尾占一个字符，共6个字符；但是你char s[5]={'a','b','c','d','e'};每个字母是一个字符，也就是5个字符

二、读程序题,写出程序运行的结果。

1、

#include <stdio.h>

void main()

{ int \*p1,\*p2,\*p;

int a=5,b=8;

p1=&a; p2=&b;

if(a<b) { p=p1; p1=p2; p2=p;}

printf("%d,%d\n",\*p1,\*p2);//8,5

printf("%d,%d\n",a,b);//8,5

}

3、

void main()

{ int a[]={2,4,6,8,10};

int y=1,x,\*p;

p=&a[1];

for(x=0;x<3;x++) y+=\*(p+x);//1+4+6+8=19

printf("y=%d\n",y);//13

}

8、

void main()

{ char s[]="ABCD",\*p;

for(p=s+1;p<s+4;p++)

printf("%s\n",p);//BCD

}

// BCD

CD

D

10、

#include <stdio.h>

#include <string.h>

void main()

{ char b1[8]="abcdefg",b2[8],\*pb=b1+3;//d

while (--pb>=b1) strcpy(b2,pb);//c>=a cba

printf("%d\n",strlen(b2));//cba

}

//7

//3

14、

void main( )

{ char \*s="12134211";

int v[4]={0,0,0,0} ;

int k,i;

for(k=0;s[k];k++)//

{ switch(s[k])

{ case '1':i=0;

case '2':i=1;

case '3':i=2;

case '4':i=3;

}

v[i]++;//0008 8

}

for(k=0;k<4;k++) printf("%d",v[k]);//0008

}

17、

void main()

{ char b[ ]="ABCDEFG";

char \*chp=&b[7];

while(--chp>&b[0])

putchar(\*chp);//GFEDCB

putchar('\n');

}

21、

#define PR(ar) printf("%d",ar)

void main( )

{ int j ,a[ ]={1,3,5,7,9,11,13,15},\*p=a+5;//11

for(j=3;j;j--)

{ switch(j)

{ case 1:

case 2:PR(\*p++);break;

case 3:PR(\*(--p));

}

}

}

//9911