C语言知识点整理

1. 数据类型
2. int 4字节 char 1字节 float 4字节 double 8字节

c=getchar(),或者写getchar()也可以，//，/\*\*/

有符号数的存储，若为负的，那么符号位之外取反然后+1，若为正的，那么不变-》无符号数

%4.7f自动补全

转义字符 \n \t \\

\1-3位八进制数

\x1-2位十六进制数，这两个表示字符

注意对于除法的运算，

对于%必须要是整数，a%b=c的符号跟着a走

对于c语言中，非0的东西即为真

Pow(a,b)表示a^b

1. 逻辑运算符表

（1）[]取数组元素，（），->指向结构体成员(比如struct stu{int jzy} struct stu \*JZY,JZY->jzy),.

(2) ! ++ -- - = （类型）强制类型转换， \*指针取值 & 取地址 sizeof

++b先加减后运算 b++先运算后加减

(3)\* / %

(4)+ -

（5）== ！=

（5）三目运算符

(6) && 》||

对于&&如果两个中间有一个是假的，那么剩下来的那个一部做了

同时，如果||中间的有一个是真，那么生下来的是不做

(7)赋值运算

（8），运算符，表示每一个表达式一次执行，最后一个表达式的值为该逗号表达式的值

从右往左

条件表达式 （表达式1）？1真 执行二：否则执行3

数学库fabs,abs cos sin pow(n,3)

1. 输入，输出，注意对于输入的时候要写上&

%c %d %f (对于输出小鼠的时候%x.yf表示有x位，左对齐，保留了y位小数)

printf(“%s”,a)(此时是输出字符串，特殊考虑)

1. 判断语句

对于流程图 圆框 起始框结束框 平四 输入输出框 方形 处理框 菱形 判断框

第一中形式If()

Else if()

Else

第二种形式 switch(x)

Case 1: …;break;

Case 2: …;break;

Default: …;break;注意加上break，否则会一直执行下去,直到第一个break;

1. 循环

For( 起始条件;每一次操作前要判断的东西，如果是真的，那么继续 ; i++（如果省略那么认为是i++） )

While(条件1为真的){}

Do()while(条件为真)

注意程序执行的次数

例：辗转相除法求最大公约数

int m,n,temp;

 while(temp!=0)

 {

 temp=m%n;

 m=n;

 n=temp;

 }

 printf("%d",n);

loop：语句标号 goto loop

例：二分查找

int a[1000];

 int low=0,mid,high=1000-1;

 while(low<=high)

 {

 mid=(low+high)/2;

 if(a[mid]==1)

 {

 printf("%d",mid);

 break;

 }

 else if(a[mid]>target)

 {

 high=mid-1;

 }

 else

 low=mid+1;

 }

1. 数组

Int a[5],则&a[0]就是a所代表的地址，

例：插入排序 于 冒泡排序

for(i=0;i<5;i++)

 {

 n=i;

 for(j=i+1;j<5;j++)

 {

 if(a[n]>a[j])

 {

 n=j;

 }

 }

 if(a[n]!=a[i])

 {

 int temp=a[i];

 a[i]=a[n];

 a[n]=temp;

 }

 }

 for(int i=0;i<5;i++)

 {

 for(int j=i+1;j<5;j++)

 {

 if(a[i]>a[i+1])

 {

 int temp=a[i];

 a[i]=a[j];

 a[j]=temp;

 }

 }

 }

矩阵乘法

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

 {

 for(int j=0;j<k;j++)

 {

 for(int l=0;l<n;l++)

 {

 c[i][j]+=a[i][l]+b[l][j];

 }

 }

 }

1. 字符串

Strlen(str2)直到第一个\0且\0不计入内

Strcmp(str1,str2)如果str1>str2 输出一个真的东西

Strcpy(str1,str2),str2直接弄到str1中间去并且覆盖原来的东西

1. 定义与初始化

Char a[]=”love” 则会在末尾自动加上一个\0

Char a[8]=”love”,则为l o v e 0 0 0 0

而对于字符串的大小就是相应的到最后的一个\0的时候

注意char \*s=”abcdefg”以及其中的s的意义

1. 字符串的输入

Scanf(%s,a)//对于这个是不可以输入空格

Gets(a)直到输入回车，注意如果后面有输入一个字符，那么需要getchar()

1. 对于函数而言

函数一定要有函数名，函数的返回类型，函数的参数类型，不一定要有形参的名字

Void swap(int n,int a[],int b[])

对于自定义函数和其余东西之间的值的传递只可以用指针，int a[],int \*a都可以

对于形参会重新开一个空间，只有在调用的时候会占内存，结束的时候释放

Static 只有在第一次调用的时候赋初值，之后调用的时候保持之前的值

二维数组的调用 int array[][4]

1. 字符串库函数

Strcpy(a,b)将b中的东西完全覆盖a

Strcat(a,b)将b中的东起全部放到a的末尾

Strcmp(a,b)如果a比b大 那么输出1，小，-1，如果相等输出0；

Puts是将最后的\0换成\n输出

 交换排序是a[i],a[j], 选择排序 选出t,交换i，t

 对于int main(int argc,char \*argv[])

1. 指针
2. 基本概念

定义 float \*p;定义了一个叫做p的float的指针类型，在使用的时候首先要将其指向一个东西，比如 p=&q,不要使用没有指向其他的东西的指针

Int a;&a表示取出a的地址，\*p表示提取出p中所指向元素的值

1. 一维数组与指针

Double a[5]={};

Double \*p=a;

P所指向的东西，又是a，又是&a[0]；而对于P++，--之类的操作就是其指向位置的移动

对于一个(\*p)[10],p+1实际上是加上指向类型的大小，p+1=10\*sizeof(int);

++,--,\* &为同一优先级的，右结合，如\*p++,\*++p都是先进行++

1. 字符数组与指针

Char a[]=”jzy”;char \*str;

Str=a;就是赋值操作

1. 二维数组

Char a[4][4];char (\*p)[4],则只要写p=a就可以了；调用的时候写p[i][j]或者是

\*（\*（p+i）+j）;

或者使用char \*p;

P=&a[0][0],或者是a[0]，\*a,在调用的时候用a[i\*m+j];

同时还有指针数组，这个比较简单

对于二维数组char a[3][3]={1,2,3,4,5,6,7,8,9};

有&a[0][0]==a[0]=\*(a+0)（注意将这里面的关于二维数组的求值全部写成标准形势），如果\*(\*(a+i)+j)中间的j超过了这一行，那么会延续到下一行中，如\*(a[0]+4)表示5

同时对于存储二维数组的时候注意是用哪一种形势去村的

对于二维数组赋初值是依次写下去，其余的东西都是0

例：自己写的（1）字符串长度统计（2）字符串赋值 （3）字符串比较

int udf\_strlen(char \*s)

{

 int len=0;

 while(\*s++)

 len++;

 return len;//this is the edition of not concluding the \0

}

char \*udf\_strcpy(char \*d,char \*s)

{

 char \*t=d;

 while(\*d++ =\*s++) ;

 return t;

}

char \*udf\_strcat(char \*d,char \*s)

{

 char \*t=d;

 while(\*d)

 d++;（不可以写成while(\*d++),这样的话会产生两个连接的字符串中间会有空格）

 while(\*d++=\*s++) ;

 return t;

}

int udf\_strcmp(char \*s,char \*t)

{

 while(\*s \*\* \*s==\*t)

 s++;t++;

 return \*s-\*t;

}

例：按照字典序排列国家

char a[10][40]; 注意，这里的建立这个东西必须要先把这些东西给存到一个二维数组里面

 for(int i=0;i<40;i++)

 {

 gets(a[i]);

 }

 char \*p[40];

 for(int i=0;i<40;i++)

 {

 p[i]=a[i];

 }//be careful to initialize the p && the a

 for(int i=0;i<40;i++)

 {

 for(int j=i+1;j<40;j++)

 {

 if(strcmp(p[i],p[j])>0)

 {

 char \*t=p[i]; 直接交换指针就可以了

 p[i]=p[j];

 p[j]=\*t;

 }

 }

 }

1. 指向函数的指针

Double f(double x)

 Double integral(double (\*f)(double),double a,double b)

1. 文件

文本 二进制 意义

R rb 只读

W wb 只写

A ab 追加

R+ r+b,rb+ 读写，打开原本的文件

W+ 读写，并且覆盖原本的东西

A+ 从原本的东西的最后开始写

FILE \*fp;

If((fp=fopen(“.txt”,”r”)!=NULL)) 注意判断要打开的文件是否是空的

Fclose(fp)

1. fscanf(fp,”%s”,name)同前面的东西，fprintf(fp…….)
2. 数据块读写 fread fwrite()

Fread(\*数据，sizeof(数据类型)，1，fp);

如fread(str+I,sizeof(char),1,fp)

1. fgetc(fp),fput(c,fp)
2. fgets=gets+stdin
3. 判断文件结束

如果结束，那么feof(fp)!=0,即为如果到达了文件的结尾，那么返回的是0

1. rewind(fp)文件位置指针回到文件的开头
2. 对于文件的依次读入

While(!feof(fp))

{

 Fscanf(…….)

}

依次读取字符

While(1)

{

C1=getc(fp1);c2=getc(fp2);

If(c1==EOF || c2==EOF)

1. 链表

1.对于struct student stu,\*p=&stu, 后缀> -> >前缀

Stu.num,(\*p).num,p->num等效；

++p->num,同++（p->num）

P++->num 同（p++）->num

2.对于结构体的定义中不可以循环定义，但是可以在内部定义指针

2.对于空间的释放函数

Malloc free

对于链表的几个容易错的东西 1.空表 2.只有一个节点 3.对于链表的第一个和最后一个节点操作

对于最后一个节点的next 应为NULL

struct student

{

 int num;

 float score;

 char name[20];

 struct student \*next;

};

struct student \*creater()

{

 int n;

 struct student \*head,\*p,\*q;

 scanf("%d",&n);

 head=NULL;

 while(n>=1)

 {

 scanf("%d%f",p->num,p->score);

 gets(p->name);

 if(head==NULL)

 {

 head=p;

 q=p;

 p->next=NULL;

 }

 else

 {

 q->next=p;

 q=p;

 }

 n--;

 }

 if(q!=NULL)

 {

 q->next=NULL;

 }

 return head;

}

struct student \*creatl()

{

 struct student \*head,\*p;

 int n;

 scanf("%d",&n);

 while(n>=1)

 {

 scanf("%d%f",p->num,p->score);

 gets(p->name);

 p->next=head;

 head=p;

 n--;

 }

 return head;

}

struct student \*insertlist(struct student \*head,struct student \*q)

{

 struct student \*p;

 if(head==NULL)

 {

 head=q;

 return head;

 }

 if(head->num>q->num)

 {

 q->next=head;

 head=q;

 return head;

 }

 p=head;

 while(p->next && p->next->num<q->num)

 {

 p=p->next;

 }

 q->next=p->next;

 p->next=q;

 return head;

}

struct student \*insertlist2(struct student \*head, int k,struct student \*q)

{

 //we assume that k is existed in the list

 struct student \*p;

 if(head==NULL)

 {

 head=q;

 return head;

 }

 if(k==0)

 {

 q->next=head;

 head=q;

 return head;

 }

 int i=0;

 while(p->next !=NULL)

 {

 if(i==k-1)

 {

 i=-1;

 break;

 }

 p=p->next;

 i++;

 }

 if(i!=0)

 {

 ...

 return head;

 }

 else

 {

 q->next=p->next;

 p->next=q;

 return head;

 }

}

struct student \*erase(struct student \*head,int num)

{

 struct student \*p;

 if(head) return head;

 if(head->num==num)

 {

 p=head;

 head=p->next;

 free(p);

 return head;

 }

 q=p=head;

 while(p && p->num!=num)

 {

 q=p;

 p=p->next;

 }

 if(p)

 {

 q->next=p->next;

 free(p);

 return head;

 }

 else

 {

 ...

 }

}

void sortnum(struct student \*head)

{

 struct student \*p,\*max,\*other;

 int tnum;float tscore;char tname[20];

 for(p=head;p!=NULL;p=p->next)

 {

 max=p;

 for(other=head;other!=NULL;other=other->next)

 {

 if(other->score<max->score)

 {

 max=other;

 }

 }

 if(max!=p)

 {

 tnum=max->num;

 max->num=p->num;

 p->num=tnum;

 strcpy(tname,max->name);

 strcpy(max->name,p->name);

 strcpy(p->name,tname);

 }

 }

}

销毁链表

node \*listdestroy(node \*head)

{

 node \*p=head,\*temp;

 while(p!=NULL)

 {

 temp=p;

 p=p->next;

 free(temp);

 }

 return NULL;

}对于链表的练习就差不多了

实际上就是对于head的特殊处理，以及注意画图就可以了。

原地逆序

Link \*reverselink(link \*H)

{

 If(H==NULL || h->next==NULL)

 Return H;

 Link \*temp,\*p=H,\*newH=NULL;

 While(p!=NULL)

 {
 temp=p->next;p->next=newH;

 newH=p;p=temp;

 }

Return newH;

}

1. 对于for(int i=0;i<100;i++)中间的i++是在每一次循环的最后面的时候去做
2. 对于# define SWAP(x,y)temp=x;x=y;y=tmep;

下面进行执行的时候也是进行循环

1. 结构体可以直接赋值

Struct student stu1,stu2;

Stu2=stu1;

1. switch语句而言如果有了一个满足要求的东西就继续执行生下来的东西直到结束。
2. 对于int (\*p)[4],其中p是一个具有4个元素的一维数组的执政

Int \*p[4] 包含n个元素的数组，p的元素类型是指向int的执政

1. 对于double t=2,t的值为2.0
2. 对于三角形的判定，就老老实实的写出所有的可能的条件
3. 注意每一次的输出
4. 对于FILE \*fp;

If((fp=fopen(“.txt”,”w+”)==NULL)

{

 Printf(“illegeal!”);

Return 0;

}

1. 对于二分查找

while(low<mid)

 {

 low=mid;

 high=mid;

 }

1. 关于标识符：

有效字符：只可以由字母，数字，下化纤组成，且以字母，下滑开头

有效： 至少前面的8个字符有效

对于大小写敏感

1. 字符串存储的时后自动加上一个\0,占用一个字节
2. 对于8进制的转义字符必须三个字符，如\78就不对
3. 逻辑运算符&& ||的优先级较后，且&&>||
4. ++>\* &>\*/>+->&& ||>=
5. For 是先进行判断再进行程序体里面的操作，i++是在程序结束的时候进行
6. Do-while是在末尾的时候进行判断
7. 函数的构成：参数列表，函数体，返回值类型，函数名四个部分
8. 注意初始化的时候，for(i=1,k=0;i<=100;i++)这个东西
9. 注意对于数组而言是从0开始，因此对于取值的时候不要看错
10. C语言程序不都是由函数和变量组成的
11. Switch中break和continure的功能不一样
12. 注意输出的时候检查每一步的输出操作和个数
13. 注意对于带参量的宏定义

#define S1(a,b) a\*b

S=S1(x+y,x-y)=x+y\*x-y;

1. 计算机自动处理的基础是存储程序
2. C语言的数据类型可以使得编译器确定在内存中如何存储一个特定的值，并且可以对该数值执行什么样的操作
3. 用户标识符不能与保留字同名。
4. %前面的操作数必须是整形或字符型
5. 如果是

void swap(int \*x, int \*y)

{

 int \*t;

 t=x,x=y,y=t;

}

int main()

{

 int a=3,b=4;

 int \*p=&a,\*q=&b;

 swap(p,q);

 printf("%d %d",a,b);

 return 0;

}

那么只是交换地址的话，那么并不可以达成交换的效果

1. Strlen(str)是统计不包含\0的字符长度
2. 函数定义具有原型声明的作用，若未指定返回值类型，那么是int, 若返回值是空，那么表示没有返回值，圆括号里面写成void表示没有参数
3. 注意冒泡排序是两两之间进行比较，0-n,0-n-i+1;
4. 指针是一种存储变量地址的变量，一半情况下指针也可以初始化，若赋值NULL则指向0x0这个地址，
5. Char \*ps=”0123”,那么sizeof(ps)=8,sizeof(\*ps)=1
6. 预处理命令行不一定是所有程序之前，他不是c语言本身的组成部分
7. Char a[]=”it is mine”,那么只可以存放10字符
8. 所有的字符型变量中只可以存储一个字符
9. 流程图





