判断闰年:1. 被4整除但不能被100整除 2. 能被400整除

整型:1 int 2或4字节 2. short int 2字节 3. long int (long)4字节 4. long long int 8字节

有无符号:signed unsigned

'a-'A'=32

```
#include<stdio.h>
int main(){
    char aa;
    scanf("%c",&aa);
    printf("%c",aa+1);
```

浮点型:1. float 精确到小数点后6位 2. double 精确到小数点后15位 3. long double 精确到小数点后18位左右

%要求操作数为整数

i=3; j=++i j=4; j=i++ j=3;

强制转化 (double) (a) 将a转化为double类型

%g去掉小数点后面的0

scanf不能输出空格回车tab等

putchar可以输出空格和回车

getchar接收字符(吃回车)

算数运算符>关系运算符>赋值运算符

关系运算符中 带>或者<大于==和!=

三目运算符 =(a>b)?a;b; 如果a>b则条件表达式=a,反之则=b

switch(字符型){

case ‘A’: ;break;

case ‘B’: ;break;

...

}

循环

do while 先执行后判断;

while 先判断再执行

break 跳出循环

continue 跳过本次循环continue后面的语句直接进行下一次循环

字符串

gets只能吃空格不能吃回车

strcat(s1,s2) 将s2接到s1后面

strcpy(s1,s2) 将s2复制到s1

strncpy(s1,s2,n) 将s2中前面n个字符复制到s1中去

strcmp(s1,s2) 比较s1与s2字符串的ASC码值若相同返回0

结构体&指针&链表

#define 宏定义 #define M struct Student;

> 将M替代struct Student

sizeof 计算字节数

```
1  char a=1
2  char*b=8
3  short int c=2
4  int d=4
5  unsigned int e=4
6  float f=4
7  double g=8
8  long h=8
9  long long i=8
10 unsiged long j=8
```

> sizeof 计算字节数 上述图片为字节数,其中结构体用malloc开空间时为了计算最大空间用sizeof(struct Student)来开空间,即
>
> (struct Student*)malloc(sizeof(struct Student))

链表:

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

typedef struct node {
    int data;
    struct node *next;
} Node;
Node* CreatList() {//创建链表
```

```
    int n;
    scanf("%d", &n);
    Node *p1, *p2, *head;
    p1 = p2 = (Node*)malloc(sizeof(Node));
    head = NULL;
    for (int i = 0; i < n; i++) {
        if (i == 0) head = p1;
        else p2->next = p1;
        p2 = p1;
        p1 = (Node*)malloc(sizeof(Node));
        scanf("%d", &p2->data);
    }
    p2->next = NULL;
    return head;
}
void ShowList(Node* s) {//输出链表
    Node *p;
    p = s;
    for (; p->next != NULL;) {
        printf("%d ", p->data);
        p = p->next;
    }
    printf("%d \n", p->data);
}
Node* ReverseList(Node* s) {//转置链表(头置法)
//  Node *p, *q;
//  p = s->next;
//  s->next = NULL;
//  for (int i=0; p->next != NULL;i++) {
//      q = p;
//      p = p->next;
//      q->next = s->next;
//      s->next = q;
//  }
//  q = p;
//  p = p->next;
//  q->next = s->next;
//  s->next = q;
//  return s;
 Node *p, *q, *r;
    p = s;
    q=r=NULL;
    while(p)
    {
        q = p->next;
        p->next = r;
        r = p;
        p = q;
    }
    return r;
}
int main(void) {
    Node *phead;
    phead = CreatList();
    printf("链表逆置前的数据:\n");
    ShowList(phead);
    phead = ReverseList(phead);
    printf("链表逆置后的数据:\n");
```

```
    ShowList(phead);
    return 0;
}
```

文件

FILE 储存文件信息

FILE* 指向文件的指针类型变量

fopen(文件名,使用文件的方式)-打开文件

使用文件方式:

> r 只读(输入) 文本文件: 转换回车-遇到'\n'转化为'\r','\n'
>
> w 只写(输出) 文本文件: 转换回车
>
> a 追加(添加) 文本文件: 转换回车
>
> rb 只读(输入) 二进制文件:不转换回车
>
> wb 只写(输出) 二进制文件:不转换回车
>
> ab 追加(添加) 二进制文件:不转换回车

fclose(文件指针)-关闭文件

```
FILE* fp;
fopen("a1","r");
fclose(fp);
```

fgets(文件指针)从fp指向的文件读入一个字符 读取失败返回EOF

fputs(字符,文件指针)将该字符存入文件

```
#include<stdio.h>
int main(){
   FILE *fp;
   if(fp=fopen(文件名,"w")==NULL){
        printf("cannot open this file");
        exit(0);//终止程序
   }else{
        ch=getchar();
        while(ch!='#'){
              fputs(ch,fp);
              putchar(ch);
              ch=getchar();
        }
        fclose(fp);
        putchar(10);//向屏幕输入一个换行符
   }
}
```