汇编实验作业

——320150938841 陈冠斌

目录

[自编程序 1](#_Toc483032055)

[I.求质数(n以内质数) 1](#_Toc483032056)

[II.求快速幂(a^b mod c) 5](#_Toc483032057)

[III.全排序 8](#_Toc483032058)

# 自编程序

——算法如何在汇编中编写

——汇编的限制和优势

限制：所需变量大，编写复杂且调试复杂，很容易出错，且很难改错

优势：可以运用汇编节省时间(c程序通过某个固定的方法直译成汇编，然后直译成机器语言再执行，有些操作重复或冗长)

## I.求质数(n以内质数)

Code：

;Problem:get prime which is lower than 10000

data segment

radix dw 0ah ;pay attention! 0ah , not 10h

n dw 100 ;if no input , n=...

vis db 0,0,10000 dup(1) ;1:prime 0:not prime ;vis[0]=false,vis[1]=false

prime dw 10000 dup(?) ;10000 in [0~2^16=65536),thus can store

mess1 db 'Please input n(n<=10000): $'

mess2 db 'Prime lower than n: $'

data ends

stack segment

db 20 dup(?)

stack ends

code segment

assume cs:code,ds:data,ss:stack

start:

mov ax,data

mov ds,ax

;output mess1

mov ah,9

mov dx,offset mess1

int 21h

;input n

mov bx,offset n

mov word ptr ds:[bx],0

mov ch,0

input:mov ah,1

int 21h

cmp al,0dh

jz init

mov cl,al

mov bx,offset n

mov ax,ds:[bx]

mov bx,offset radix

mul word ptr ds:[bx]

add ax,cx

sub ax,30h

mov bx,offset n

mov word ptr [bx],ax

jmp input

;initialization

init:mov bx,offset n

mov cx,[bx]

dec cx

mov si,1 ;i(current prime)

mov di,0 ;ans\*2(dw)

A:inc si

mov bx,offset vis

cmp byte ptr [bx+si],1

jnz B

;si is a prime

mov bx,offset prime

mov [bx+di],si

add di,2

B:push cx

mov cx,0

;if i\*zhi[j]>n then break

C:mov bx,offset prime

add bx,cx

mov ax,si

mul word ptr [bx] ;dx\_ax:dx=0

mov bx,offset n

cmp ax,[bx]

ja D

;vis[i\*zhi[j]]=false

mov bx,offset vis

add bx,ax

mov al,0

mov [bx],al

;if i%zhi[j]=0 then break

mov bx,offset prime

add bx,cx

mov dx,0

mov ax,si

div word ptr [bx]

cmp dx,0

jz D

add cx,2

cmp cx,di

jnz C

D:pop cx

loop A

;output mess2

mov dx,offset mess2

mov ah,9

int 21h

;output

mov bx,offset prime

mov ax,stack

mov ss,ax

mov ax,20

mov sp,ax

E:mov dx,0

mov ax,ds:[bx]

div word ptr ds:[0]

inc cx

push dx ;dx<10 00dl

mov ds:[bx],ax

cmp ax,0

jnz E

mov ah,2

F:pop dx

add dx,30h

int 21h

loop F

mov dl,' '

int 21h

sub di,2

jz G

add bx,2

jmp E

G:mov ah,4ch

int 21h

code ends

end start

Analysis：

1.采用Eratosthenes筛法，时间复杂度O(n)

2.程序包括：

A：输入一个十进制的数并保存在一个内存单元中

B：每次一个乘法，两个除法运算，共n次

C：以十进制形式输出

3.C代码

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

#include <stdbool.h>

#include <memory.h>

int main()

{

bool \*vis=(bool \*) malloc (sizeof(bool)\*100000000);

long \*zhi=(long \*) malloc (sizeof(long)\*10000000);

long n=1000,ans,i,j;

memset(vis,true,sizeof(vis)\*(n+1));

vis[2]=true;

ans=0;

for (i=2;i<=n;i++)

{

if (vis[i])

{

ans++;

zhi[ans]=i;

}

for (j=1;j<=ans;j++)

{

if (i\*zhi[j]>n)

break;

vis[i\*zhi[j]]=false;

if (i%zhi[j]==0)

break;

}

}

printf("%ld\n",ans);

// for (i=1;i<=ans;i++)

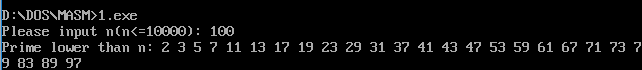
// printf("%ld ",zhi[i]);

printf("\n");

return 0;

}

Result：



## II.求快速幂(a^b mod c)

Code：

;Quickpow

data segment

;dw [0,65536)

dw 2,20,1000,1

;x,y,z,result result=x^y mod z

;x,y,z<65536 result<65536

vis dw 0

dw 0ah

data ends

stack segment

db 20 dup(?)

stack ends

code segment

assume cs:code,ds:data,ss:stack

start:mov ax,data

mov ds,ax

mov cx,ds:[2] ;2=1(place)\*2(length)

A:mov dx,cx

and dx,1

jz B

mov ax,ds:[6]

mul word ptr ds:[0] ;result\*a%c

div word ptr ds:[4]

mov ds:[6],dx

B:mov ax,ds:[0]

mul word ptr ds:[0] ;a\*a%c

div word ptr ds:[4]

mov ds:[0],dx

shr cx,1 ;allow: shr xx,1 / shr xx,cl

jnz A

;output the number from ds:[6]

mov bx,ds:[6]

mov ch,4

C:mov cl,4

rol bx,cl

mov dx,bx

and dx,0fh

add dl,30h

cmp dl,3ah

jl D

add dl,7h

D:cmp dl,30h

jnz output

cmp word ptr ds:[8],0

jnz output

mov dx,1

mov ds:[8],dx

dec ch

jnz C

output:mov ah,2

int 21h

dec ch

jnz C

;mov ah,2

mov dl,'H'

int 21h

mov dl,0dh

int 21h

mov dl,0ah

int 21h

;output from ds:[6], decimal

mov ax,stack

mov ss,ax

mov ax,20

mov sp,ax

mov cx,0

E:mov dx,0

mov ax,ds:[6]

div word ptr ds:[10]

inc cx

push dx ;dx<10 00dl

mov ds:[6],ax

cmp ax,0

jnz E

mov ah,2

F:pop dx

add dx,30h

int 21h

loop F

;mov ah,2

mov dl,'D'

int 21h

mov ah,4ch

int 21h

code ends

end start

Analysis：

1.时间复杂度O(log(b))

2.程序包括：

A：每次循环两个乘法，两个除法，大约log(b)次循环

B：以十六进制输出数，首位0不输出

C：以十进制输出数

3.C代码

#include <stdio.h>

#include <stdlib.h>

int main()

{

long a,b,result=1;

//a^b

scanf("%ld%ld",&a,&b);

while (b)

{

if ((b & 1)==1)

result=result\*a;

a=a\*a;

b>>=1;

}

printf("%ld\n",result);

return 0;

}

Result：



## III.全排序

Code：

;QuanPaiLie

data segment

num db 10 dup(?)

mess1 db 'Please input n(n<=8): $'

mess2 db 'QuanPaiLie: ',0dh,0ah,'$'

data ends

stack segment

db 20 dup(?)

stack ends

code segment

assume cs:code,ds:data,ss:stack

start:mov ax,data

mov ds,ax

mov ax,stack

mov ss,ax

mov ax,20

mov sp,ax

;output mess1

mov ah,9

mov dx,offset mess1

int 21h

;input n(n<10)

mov ah,1

int 21h

mov cl,al

mov ch,0

sub cx,30h

mov dx,cx

push cx

push cx

;output '\n'

mov ah,2

mov dl,0dh

int 21h

mov dl,0ah

int 21h

;output mess2

mov ah,9

mov dx,offset mess2

int 21h

;initialization:n number(s)

mov bx,0

A:inc bx

mov word ptr [bx],bx ;begin from xx:0001

loop A

;total

pop cx

mov ax,1

B:mul cx

loop B

mov cx,ax

;work

pop dx

C:

;output n number(s)

push cx

mov bx,0

mov ah,2

mov cx,dx

push dx

D:inc bx

mov dl,[bx]

add dl,30h

int 21h

loop D

mov dl,0dh

int 21h

mov dl,0ah

int 21h

pop dx

;;change kth condition into (k+1)th condition

;find a position i , st. a[i]<a[i+1]

mov si,dx

E:dec si

mov al,ds:[si] ;?

cmp al,ds:[si+1] ;?

;cmp byte ptr ds:[si],ds:[si+1]

ja E

;find a biggest position j , st. a[i]<a[j]

mov di,dx

inc di

F:dec di

mov al,ds:[si] ;?

cmp al,ds:[di]

;cmp byte ptr ds:[si],ds:[di]

ja F

;change a[i],a[j]

mov al,ds:[si]

mov bl,ds:[di] ;?

mov ds:[si],bl ;?

;mov byte ptr ds:[si],ds:[di]

mov ds:[di],al

;reverse a[i+1]~a[n]

mov di,dx

inc di

G:inc si

dec di

;if si>=di , it means the finish of reverse

cmp si,di

jnb H

;change

mov al,ds:[si]

mov bl,ds:[di] ;?

mov ds:[si],bl ;?

;mov byte ptr ds:[si],ds:[di]

mov ds:[di],al

jmp G

H:pop cx

loop C

mov ah,4ch

int 21h

code ends

end start

Analysis：

1.n的全排列用有n！个，所以n不能太大。输入n小于等于8的数(n!小于65536)，直接用ah=2

,int 21h输入即可。

2.程序包括：

多次数据的查找和交换

3.C代码

#include <stdio.h>

#include <stdlib.h>

#define maxn 15

//编号为k的数列通过一系列操作改变为编号为k+1的数列

int main()

{

long j,k,n,temp,x,y,a[maxn+1];

long long i,s;

scanf("%ld",&n);

//posibility

s=1;

for (i=2;i<=n;i++)

s=s\*i;

for (i=1;i<=n;i++)

a[i]=i;

for (k=1;k<=n;k++)

printf("%ld ",a[k]);

printf("\n");

for (i=1;i<s;i++)

{

//4 8 7 6 5 3 2 1

//5 8 7 6 4 3 2 1

//5 1 2 3 4 6 7 8

//从尾到头，找到第一个下降的a[j]

for (j=n-1;j>=1;j--)

if (a[j]<a[j+1])

break;

//a[j]：从尾到a[j],找到第一个比a[j]大的数a[k]

for (k=n;k>j;k--)

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

break;

//交换a[j]和a[k]的值

temp=a[j];

a[j]=a[k];

a[k]=temp;

//数组：j+1~n reverse

x=j+1;

y=n;

while (x<y)

{

temp=a[x];

a[x]=a[y];

a[y]=temp;

x++;

y--;

}

for (k=1;k<=n;k++)

printf("%ld ",a[k]);

printf("\n");

}

return 0;

}

Result：

