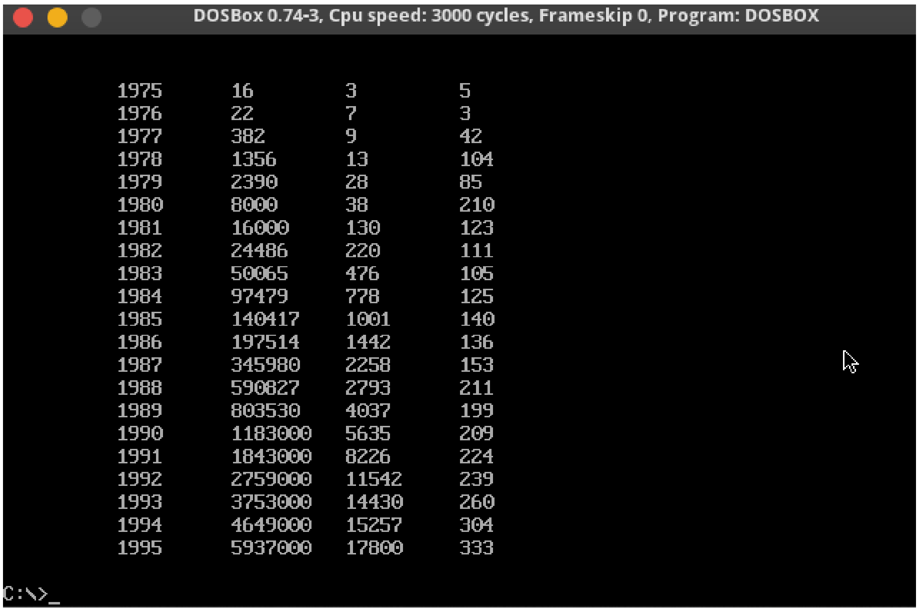
课程设计1

代码

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

data segment
        ;0,0
        db '1975','1976','1977','1978','1979','1980','1981','1982','1983'
        db '1984','1985','1986','1987','1988','1989','1990','1991','1992'
        db  '1993','1994','1995'
        ;以上是表示21年的21个字符串 year

        ;84,54h
        dd 16,22,382,1356,2390,8000,16000,24486,50065,97479,140417,197514
        dd 345980,590827,803530,1183000,1843000,2759000,3753000,4649000,5937000
        ;以上是表示21年公司总收入的21个dword数据   sum

        ;168,a8
        dw 3,7,9,13,28,38,130,220,476,778,1001,1442,2258,2793,4037,5635,8226
        dw  11542,14430,15257,17800
data ends



stack segment
    db 128 dup(0)
stack ends


table segment   ;0123456789ABCDEF
    db  21 dup ('year summ ne ?? ')
table ends

string segment ;0123456789ABCDEF
    db  21 dup (0000000000000000)
string ends
```

```


code segment
start:
            mov ax,stack;stack
            mov ss,ax
            mov sp,128

            call clear_screen;清空屏幕

            call init_reg;初始化寄存器组，把数据导入

            call input_table;把数据格式化存储

            call output_table;输出格式化数据

            mov ax,4C00H
            int 21H

;================================;output_table
output_table:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

    output_table_bg:
            ;分别调用显示子程序
            call print_year;年
            call print_income;总收入
            call print_employee;员工数量
            call print_avg;人均收入

    output_table_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret
    ;=============================================
    print_avg:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov ax,table
            mov ds,ax
            mov ax,string
            mov es,ax

            mov si,0

        print_avg_bg:

            mov cx,21
            mov di,160*3+2*40;在屏幕中的显示位置
        pa_lp1:
            push cx

            mov ax,ds:[si+13]
            mov dx,0
            mov cx,10

            call transfer;改进后的子程序3

            add si,16
            add di,160
            pop cx
            loop pa_lp1

            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret

    ;=============================================
    print_employee:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov ax,table
            mov ds,ax
            mov ax,string
            mov es,ax

            mov si,0

        print_employee_bg:
            mov cx,21
            mov di,160*3+2*30

        peb_lp1:
            push cx

            mov ax,ds:[si+10]
            mov dx,0
            mov cx,10

            call transfer

            add si,16
            add di,160
            pop cx
            loop peb_lp1

        print_employee_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret

    ;=============================================
    print_income:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov ax,table
            mov ds,ax
            mov ax,string
            mov es,ax

            mov si,0


        print_income_bg:

            mov cx,21
            mov di,160*3+2*20
        pib_lp1:
            push cx

            mov ax,ds:[si+5]
            mov dx,ds:[si+7]
            mov cx,10


            call transfer

            add si,16
            ;add di,16
            add di,160
            pop cx
            loop pib_lp1


        print_income_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret

        ;==============================
        transfer:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov bx,15

        transfer_bg:    ;判断是否大于一个字节
            mov cx,dx
            jcxz short_div;不大于一个字节

            mov cx,10
            push ax
            mov bp,sp
            call divdw
            add sp,2

            add cl,30h
            mov es:[bx],cl

            dec bx
            jmp transfer_bg
        sn:
            call show_number;输出结果

        transfer_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret
        ;====================
        show_number:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov ax,string
            mov ds,ax
            mov ax,0b800h
            mov es,ax


        sn_lp1:
            mov cx,0
            mov cl,ds:[bx]
            jcxz show_number_end
            mov ch,00000111b
            mov es:[di],cx

            inc bx
            add di,2

            jmp sn_lp1

        show_number_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret

        ;===================
        divdw:

        mov ax,dx
        mov dx,0

        div cx
        push ax
        mov ax,ss:[bp+0]
        div cx

        mov cx,dx
        pop dx

        ret

        ;===================
        short_div:
        mov cx,10
        div cx
        add dl,30h
        mov es:[bx],dl
        mov cx,ax
        jcxz sn
        dec bx
        mov dx,0
        jmp short_div

    ;=============================================
    print_year:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            ;0123456789ABCDEF
            ;year summ ne ??

        print_year_bg:
            mov ax,table
            mov ds,ax
            mov ax,0b800h
            mov es,ax
            mov si,0
            mov di,160*3+2*10

            mov cx,21
        pyb_lp1:
            push cx
            mov cx,4
            mov bx,0
        pyb_lp2:
            mov al,ds:[si+bx]
            ;mov ah,00000111b
            mov ah,00000111b
            push bx
            add bx,bx
            mov es:[di+bx],ax
            pop bx

            inc bx
            loop pyb_lp2

            add si,16
            add di,160
            pop cx
            loop pyb_lp1

        print_year_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret

;=================================;input_table
input_table:
            push ax
            push bx
            push cx
            push dx
            push es
            push ds
            push di
            push si

            mov si,0
            mov di,0
            mov bx,21*4*2

            mov cx,21


                ;0123456789ABCDEF
                ;year summ ne ??
    input_table_bg:
            ;year
            push ds:[si+0]
            pop es:[di+0]
            push ds:[si+2]
            pop es:[di+2]
            ;income
            mov ax,ds:[si+21*4+0]
            mov dx,ds:[si+21*4+2]
            mov es:[di+5],ax
            mov es:[di+7],dx
            ;employee
            push ds:[bx]
            pop es:[di+10]
            ;avg
            div word ptr ds:[bx]
            mov es:[di+13],ax

            add si,4
            add di,16
            add bx,2

            loop input_table_bg
    input_table_end:
            pop si
            pop di
            pop ds
            pop es
            pop dx
            pop cx
            pop bx
            pop ax
            ret



;==============================初始化寄存器组，把数据导入
init_reg:
            mov bx,data;data in
            mov ds,bx
            mov bx,table;data out
            mov es,bx
            ret

;===================================清空屏幕
clear_screen:
            mov bx,0b800h
            mov es,bx

            mov bx,0
            mov dx,0000h
            mov cx,2000

    clearScreen:
                    mov es:[bx],dx
                    add bx,2

                    loop clearScreen

                    ret
code ends

end start
```

截图


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```


          1975      16        3         5
          1976      22        7         3
          1977      382       9         42
          1978      1356      13        104
          1979      2390      28        85
          1980      8000      38        210
          1981      16000     130       123
          1982      24486     220       111
          1983      50065     476       105
          1984      97479     778       125
          1985      140417    1001      140
          1986      197514    1442      136
          1987      345980    2258      153
          1988      590827    2793      211
          1989      803530    4037      199
          1990      1183000   5635      209
          1991      1843000   8226      224
          1992      2759000   11542     239
          1993      3753000   14430     260
          1994      4649000   15257     304
          1995      5937000   17800     333

C:\>_
```