

课程设计1

代码

```
1  assume cs:code,ds:data,ss:stack
2
3  data segment
4      ;0,0
5      db
6      '1975','1976','1977','1978','1979','1980','1981','19
7      82','1983'
8      db
9      '1984','1985','1986','1987','1988','1989','1990','19
10     91','1992'
11     db  '1993','1994','1995'
12     ;以上是表示21年的21个字符串 year
13
14     ;84,54h
15     dd
16     16,22,382,1356,2390,8000,16000,24486,50065,97479,140
17     417,197514
18     dd
19     345980,590827,803530,1183000,1843000,2759000,3753000
20     ,4649000,5937000
21     ;以上是表示21年公司总收入的21个dword数据 sum
22
23     ;168,a8
24     dw
25     3,7,9,13,28,38,130,220,476,778,1001,1442,2258,2793,4
26     037,5635,8226
27     dw 11542,14430,15257,17800
28 data ends
29
30
31 stack segment
32     db 128 dup(0)
33 stack ends
34
35 table segment ;0123456789ABCDEF
36     db 21 dup ('year summ ne ?? ')
37 table ends
38
39 string segment ;0123456789ABCDEF
40     db 21 dup (0000000000000000)
41 string ends
```

```

34
35
36 code segment
37 start:
38     mov ax,stack;stack
39     mov ss,ax
40     mov sp,128
41
42     call clear_screen;清空屏幕
43
44     call init_reg;初始化寄存器组，把数据导入
45
46     call input_table;把数据格式化存储
47
48     call output_table;输出格式化数据
49
50     mov ax,4C00H
51     int 21H
52
53     ;=====;output_table
54 output_table:
55     push ax
56     push bx
57     push cx
58     push dx
59     push es
60     push ds
61     push di
62     push si
63
64     output_table_bg:
65         ;分别调用显示子程序
66         call print_year;年
67         call print_income;总收入
68         call print_employee;员工数量
69         call print_avg;人均收入
70
71     output_table_end:
72         pop si
73         pop di
74         pop ds
75         pop es
76         pop dx
77         pop cx
78         pop bx
79         pop ax
80         ret
81     ;=====
82 print_avg:
83     push ax
84     push bx
85     push cx

```

```

86         push dx
87         push es
88         push ds
89         push di
90         push si
91
92         mov ax,table
93         mov ds,ax
94         mov ax,string
95         mov es,ax
96
97         mov si,0
98
99     print_avg_bg:
100
101         mov cx,21
102         mov di,160*3+2*40;在屏幕中的显示位置
103     pa_lp1:
104         push cx
105
106         mov ax,ds:[si+13]
107         mov dx,0
108         mov cx,10
109
110         call transfer;改进后的子程序3
111
112         add si,16
113         add di,160
114         pop cx
115         loop pa_lp1
116
117         pop si
118         pop di
119         pop ds
120         pop es
121         pop dx
122         pop cx
123         pop bx
124         pop ax
125         ret
126
127     ;=====
128     print_employee:
129         push ax
130         push bx
131         push cx
132         push dx
133         push es
134         push ds
135         push di
136         push si
137

```

```

138         mov ax,table
139         mov ds,ax
140         mov ax,string
141         mov es,ax
142
143         mov si,0
144
145     print_employee_bg:
146         mov cx,21
147         mov di,160*3+2*30
148
149     peb_lp1:
150         push cx
151
152         mov ax,ds:[si+10]
153         mov dx,0
154         mov cx,10
155
156         call transfer
157
158         add si,16
159         add di,160
160         pop cx
161         loop peb_lp1
162
163     print_employee_end:
164         pop si
165         pop di
166         pop ds
167         pop es
168         pop dx
169         pop cx
170         pop bx
171         pop ax
172         ret
173
174     ;=====
175     print_income:
176         push ax
177         push bx
178         push cx
179         push dx
180         push es
181         push ds
182         push di
183         push si
184
185         mov ax,table
186         mov ds,ax
187         mov ax,string
188         mov es,ax
189

```

```

190         mov si,0
191
192
193     print_income_bg:
194
195         mov cx,21
196         mov di,160*3+2*20
197     pib_lp1:
198         push cx
199
200         mov ax,ds:[si+5]
201         mov dx,ds:[si+7]
202         mov cx,10
203
204
205         call transfer
206
207         add si,16
208         ;add di,16
209         add di,160
210         pop cx
211         loop pib_lp1
212
213
214     print_income_end:
215         pop si
216         pop di
217         pop ds
218         pop es
219         pop dx
220         pop cx
221         pop bx
222         pop ax
223         ret
224
225     ;=====
226     transfer:
227         push ax
228         push bx
229         push cx
230         push dx
231         push es
232         push ds
233         push di
234         push si
235
236         mov bx,15
237
238     transfer_bg:     ;判断是否大于一个字节
239         mov cx,dx
240         jcxz short_div;不大于一个字节
241

```

```

242         mov cx,10
243         push ax
244         mov bp,sp
245         call divdw
246         add sp,2
247
248         add cl,30h
249         mov es:[bx],cl
250
251         dec bx
252         jmp transfer_bg
253     sn:
254         call show_number;输出结果
255
256     transfer_end:
257         pop si
258         pop di
259         pop ds
260         pop es
261         pop dx
262         pop cx
263         pop bx
264         pop ax
265         ret
266     ;=====
267     show_number:
268         push ax
269         push bx
270         push cx
271         push dx
272         push es
273         push ds
274         push di
275         push si
276
277         mov ax,string
278         mov ds,ax
279         mov ax,0b800h
280         mov es,ax
281
282
283     sn_lp1:
284         mov cx,0
285         mov cl,ds:[bx]
286         jcxz show_number_end
287         mov ch,00000111b
288         mov es:[di],cx
289
290         inc bx
291         add di,2
292
293         jmp sn_lp1

```

```

294
295     show_number_end:
296         pop si
297         pop di
298         pop ds
299         pop es
300         pop dx
301         pop cx
302         pop bx
303         pop ax
304         ret
305
306     ;=====
307     divdw:
308
309     mov ax,dx
310     mov dx,0
311
312     div cx
313     push ax
314     mov ax,ss:[bp+0]
315     div cx
316
317     mov cx,dx
318     pop dx
319
320     ret
321
322     ;=====
323     short_div:
324     mov cx,10
325     div cx
326     add dl,30h
327     mov es:[bx],dl
328     mov cx,ax
329     jcxz sn
330     dec bx
331     mov dx,0
332     jmp short_div
333
334     ;=====
335     print_year:
336         push ax
337         push bx
338         push cx
339         push dx
340         push es
341         push ds
342         push di
343         push si
344
345         ;0123456789ABCDEF

```

```

346             ;year summ ne ??
347
348     print_year_bg:
349         mov ax,table
350         mov ds,ax
351         mov ax,0b800h
352         mov es,ax
353         mov si,0
354         mov di,160*3+2*10
355
356         mov cx,21
357     pyb_lp1:
358         push cx
359         mov cx,4
360         mov bx,0
361     pyb_lp2:
362         mov al,ds:[si+bx]
363         ;mov ah,00000111b
364         mov ah,00000111b
365         push bx
366         add bx,bx
367         mov es:[di+bx],ax
368         pop bx
369
370         inc bx
371         loop pyb_lp2
372
373         add si,16
374         add di,160
375         pop cx
376         loop pyb_lp1
377
378     print_year_end:
379         pop si
380         pop di
381         pop ds
382         pop es
383         pop dx
384         pop cx
385         pop bx
386         pop ax
387         ret
388
389     ;=====;input_table
390     input_table:
391         push ax
392         push bx
393         push cx
394         push dx
395         push es
396         push ds
397         push di

```



```

398         push si
399
400         mov si,0
401         mov di,0
402         mov bx,21*4*2
403
404         mov cx,21
405
406
407         ;0123456789ABCDEF
408         ;year summ ne ??
409     input_table_bg:
410         ;year
411         push ds:[si+0]
412         pop es:[di+0]
413         push ds:[si+2]
414         pop es:[di+2]
415         ;income
416         mov ax,ds:[si+21*4+0]
417         mov dx,ds:[si+21*4+2]
418         mov es:[di+5],ax
419         mov es:[di+7],dx
420         ;employee
421         push ds:[bx]
422         pop es:[di+10]
423         ;avg
424         div word ptr ds:[bx]
425         mov es:[di+13],ax
426
427         add si,4
428         add di,16
429         add bx,2
430
431         loop input_table_bg
432     input_table_end:
433         pop si
434         pop di
435         pop ds
436         pop es
437         pop dx
438         pop cx
439         pop bx
440         pop ax
441         ret
442
443
444
445     ;=====初始化寄存器组，把数据导入
446     init_reg:
447         mov bx,data;data in
448         mov ds,bx
449         mov bx,table;data out

```

```

450             mov es,bx
451             ret
452
453 ;=====清空屏幕
454 clear_screen:
455             mov bx,0b800h
456             mov es,bx
457
458             mov bx,0
459             mov dx,0000h
460             mov cx,2000
461
462             clearScreen:
463                 mov es:[bx],dx
464                 add bx,2
465
466                 loop clearScreen
467
468             ret
469 code ends
470
471 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:\>_