

# 西安电子科技大学

## 微机原理与系统设计 课程实验报告

实验名称 第四章上机习题

人工智能 学院 1920012 班  
姓名 杨文韬 学号 18020100245  
同作者  
实验日期 2021 年 10 月 20 日

成绩

指导教师评语:

指导教师:

年 月 日

### 实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验结果与分析（图表等）
- 五、实验的收获及心得体会
- 六、实验代码

# 目录

<b>1</b>	<b>实验目的</b>	<b>1</b>
<b>2</b>	<b>实验环境</b>	<b>1</b>
<b>3</b>	<b>方案设计</b>	<b>1</b>
3.1	题目 4.47 设计方案 . . . . .	1
3.2	题目 4.53 设计方案 . . . . .	1
<b>4</b>	<b>实验结果与分析</b>	<b>3</b>
<b>5</b>	<b>实验的收获及心得</b>	<b>3</b>
<b>6</b>	<b>实验代码</b>	<b>3</b>
6.1	题目 4.47 代码 . . . . .	3
6.2	题目 4.53 代码 . . . . .	5

## 第四章上机习题

### 1 实验目的

完成第四章习题里的两个上机题目，如下

- 4.47 用同余法产生 200 个小于 256 的伪随机数，统计其中奇数的个数，并计算所有奇数的和，将奇数个数存入名为 CNT 的字节单元，和存入名为 SUMODD 的字存储单元中。用完整的段定义语句编写出实现这一功能的汇编语言源程序。
- 4.53 设有  $n$  (设为 17) 个人围坐在圆桌周围，按顺时针给他们编号  $(1, 2, \dots, n)$ ，从第 1 个人开始顺时针方向加 1 报数，当报数到  $m$  (设为 11) 时，该人出列，余下的人继续进行，直到所有人出列。编写程序模拟这一过程，求出出列人的编号顺序。

### 2 实验环境

- emu8086: 编写汇编语言程序，并进行编译和连接，生成一个可执行程序
- L<sup>A</sup>T<sub>E</sub>X: 制作封面并进行文档编写排版
- Visio 2016: 程序相关流程图绘制

### 3 方案设计

#### 3.1 题目 4.47 设计方案

设计的产生伪随机数的递归公式为

$$X_{i+1} \equiv 5X_i + 3 \pmod{256}$$

题目 4.47 设计的流程图如图 1 所示

#### 3.2 题目 4.53 设计方案

题目 4.53 设计的流程图如图 2 所示

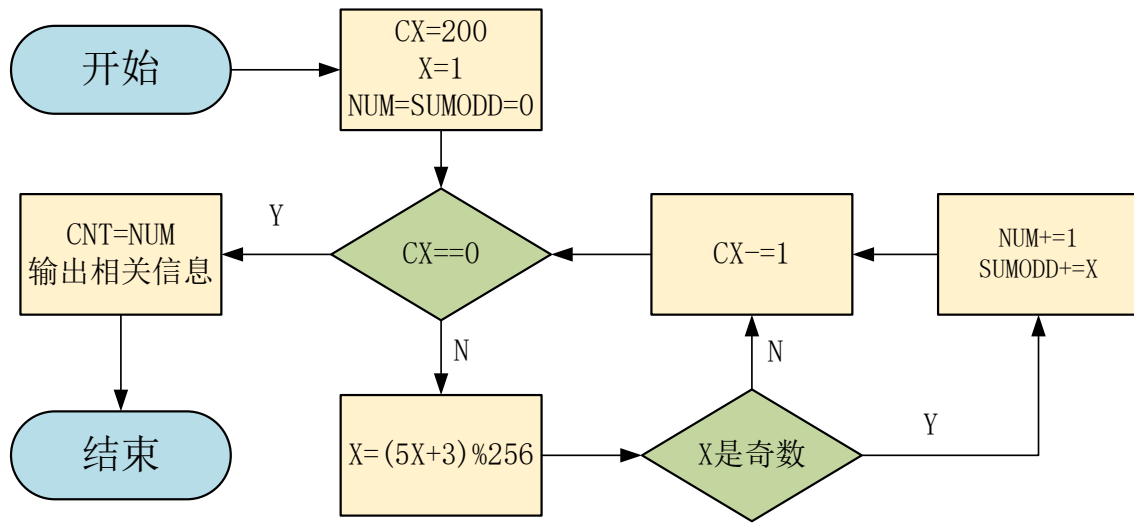


图 1: 题目 4.47 设计流程图

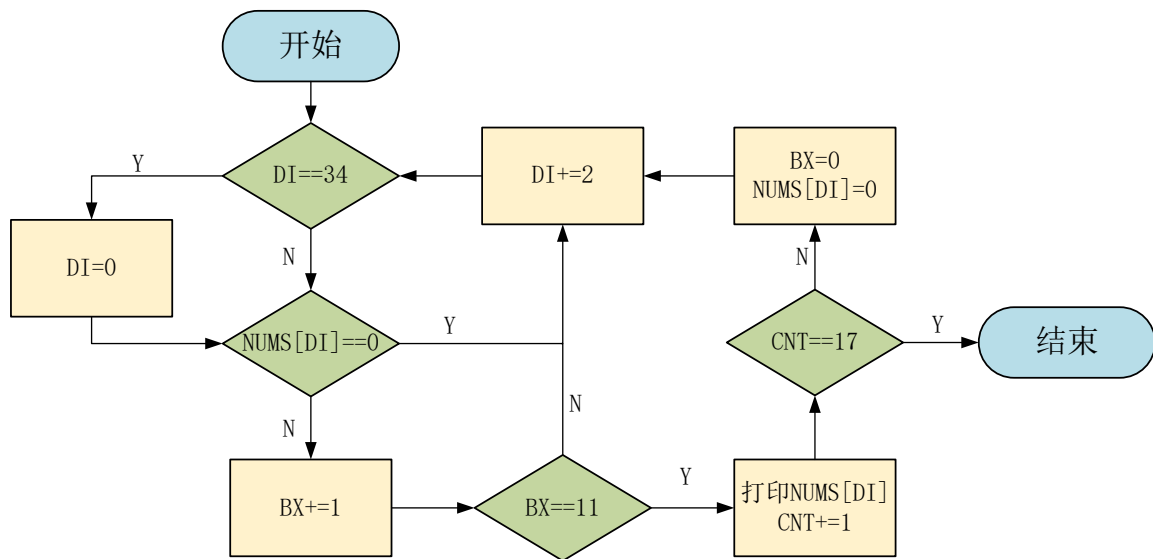


图 2: 题目 4.53 设计流程图

COUNT OF ODD NUMBERS: 100  
SUM OF ODD NUMBERS: 13256

图 3: 题目 4.47 实验结果图

## 4 实验结果与分析

题目 4.47 的输出结果显示, 产生了 100 个奇数, 它们的和是 13256, 如图 3 所示。

题目 4.53 的输出结果显示, 出列人的编号顺序为 11,5,17,13,9,7,6,8,12,16,4,2,3,15,10,14,1, 如图 4 所示。

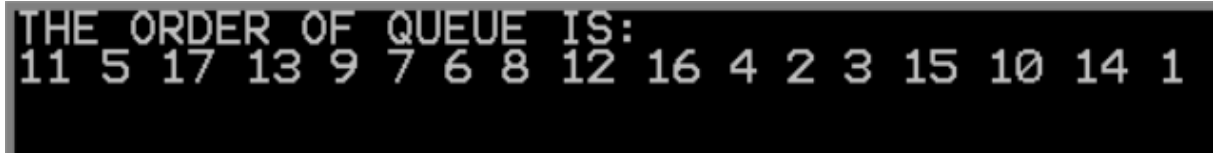


图 4: 题目 4.53 实验结果图

## 5 实验的收获及心得

通过本次实验, 我掌握了同余法生成伪随机数的方法和约瑟夫问题的汇编语言解法, 提高了汇编语言程序设计技巧, 能够较好地将思路转化为流程图, 进一步熟悉了 L<sup>A</sup>T<sub>E</sub>X 排版和 Visio 绘图技巧。但本次实验中的不足是, 我虽然能够想到时间复杂度更低的算法, 但由于汇编语言里不好实现一些数据结构而难以将其实现。

## 6 实验代码

### 6.1 题目 4.47 代码

```

1  DATA SEGMENT
2      STR1 DB 'COUNT OF ODD NUMBERS:', '$'
3      STR2 DB 13,10,'SUM OF ODD NUMBERS:', '$'
4      STR0 DB 13,10,'$'
5      CNT DW 0
6      SUMODD DW 0
7      NUM DW 0
8      X DW 1
9      A DW 5
10     B DW 3
11     MOD DW 256
12 DATA ENDS
13
14 STACK SEGMENT STACK 'STACK'
15     DW 100H DUP(?)
16 STACK ENDS
17

```

```

18 CODE SEGMENT
19     ASSUME CS:CODE,DS:DATA,SS:STACK
20 START:
21     MOV AX, DATA
22     MOV DS, AX
23     MOV AX, STACK
24     MOV SS, AX
25     MOV CX, 200
26
27 FOR1:
28     MOV AX, X
29     MUL A
30     ADD AX, B
31     DIV MOD
32     MOV AX, DX ; (A*X+B)%MOD, A=5, B=3, MOD=256
33     MOV X, AX
34     SHR AX, 1
35     JNC EVEN
36     RCL AX, 1
37     INC NUM
38     ADD SUMODD, AX
39     ; LEA SI, X
40     ; CALL PRINT
41     ; MOV DX, OFFSET STRO
42     ; MOV AH, 9
43     ; INT 21H
44 EVEN:
45     LOOP FOR1
46
47     MOV DX, OFFSET STR1
48     MOV AH, 9
49     INT 21H
50     LEA SI, NUM
51     CALL PRINT
52     MOV AL, BYTE PTR NUM
53     MOV CNT, AL
54
55     MOV DX, OFFSET STR2
56     MOV AH, 9
57     INT 21H
58     LEA SI, SUMODD
59     CALL PRINT
60
61     MOV AH, 4CH
62     INT 21H
63
64 PRINT PROC NEAR

```

```

65     PUSH AX
66     PUSH BX
67     PUSH CX
68     PUSH DX
69
70     MOV AX, [SI]
71     MOV BX, 10
72     MOV CX, 0
73 INIT:
74     XOR DX, DX
75     DIV BX ; 商 AX 余 DX
76     INC CX
77     PUSH DX
78     CMP AX, 0
79     JNZ INIT
80 OUTPUT:
81     POP DX
82     OR DX, 30H
83     MOV AH, 2
84     INT 21H
85     LOOP OUTPUT
86
87     POP DX
88     POP CX
89     POP BX
90     POP AX
91     RET
92 PRINT ENDP
93
94 CODE ENDS
95 END START

```

## 6.2 题目 4.53 代码

```

1 DATA SEGMENT
2     NUMS DW 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 0
3     STR DB 'THE ORDER OF QUEUE IS:',13,10,'$'
4     CNT DW 0
5 DATA ENDS
6
7 STACK SEGMENT STACK 'STACK'
8     DW 100H DUP(?)
9 STACK ENDS
10
11 CODE SEGMENT

```

```

12     ASSUME CS:CODE, DS:DATA, SS:STACK
13 START:
14     MOV AX, DATA
15     MOV DS, AX
16     MOV AX, STACK
17     MOV SS, AX
18
19     XOR BX, BX ; 当前报的数
20     XOR SI, SI
21     XOR DI, DI
22
23     MOV DX, OFFSET STR
24     MOV AH, 9
25     INT 21H
26
27 L0:
28     CMP DI, 34
29     JNZ L1
30     XOR DI, DI
31 L1:
32     CMP NUMS[DI], 0
33     JNZ L2
34     ADD DI, 2
35     JMP L0
36 L2:
37     INC BX
38     CMP BX, 11
39     JZ OK
40     ADD DI, 2
41     JMP L0
42 OK:
43     PUSH SI ; 打印出列顺序
44     LEA SI, NUMS[DI]
45     CALL PRINT
46     MOV DL, 32
47     MOV AH, 2
48     INT 21H
49     POP SI
50
51     INC CNT
52     CMP CNT, 17
53     JZ OVER
54     XOR BX, BX ; 报数清0
55     MOV NUMS[DI], 0
56     ADD DI, 2
57     JMP L0
58 OVER:

```



```

59
60     MOV AH, 4CH
61     INT 21H
62
63 PRINT PROC NEAR
64     PUSH AX
65     PUSH BX
66     PUSH CX
67     PUSH DX
68
69     MOV AX, [SI]
70     MOV BX, 10
71     MOV CX, 0
72 INIT:
73     XOR DX, DX
74     DIV BX ; 商 AX 余 DX
75     INC CX
76     PUSH DX
77     CMP AX, 0
78     JNZ INIT
79 OUTPUT:
80     POP DX
81     OR DX, 30H
82     MOV AH, 2
83     INT 21H
84     LOOP OUTPUT
85
86     POP DX
87     POP CX
88     POP BX
89     POP AX
90     RET
91 PRINT ENDP
92
93 CODE ENDS
94 END START

```