学号 E12214068 专业 计算机科学与技术 姓名 鲁正扬
实验日期 **2024.6.12** 教师签字 成绩

实验报告

【实验名称】 8086 宏汇编程序设计

【实验目的】

使用宏汇编设计解决程序问题。

【实验原理】

1. 宏汇编程序设计

2. 宏展开

3. 宏调用

【实验内容】

实验一：

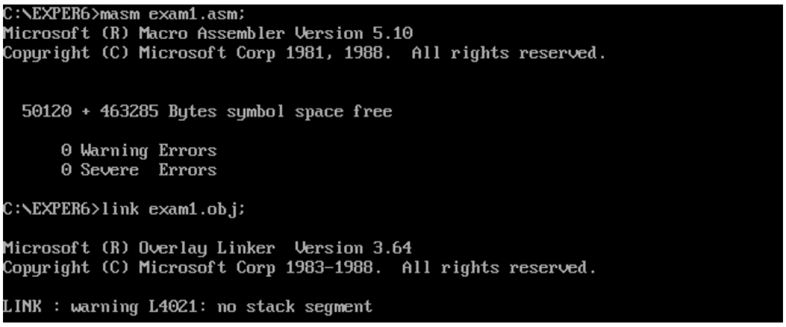
一. 实现功能：

要求：使用宏汇编实现字符串的显示；

代码：

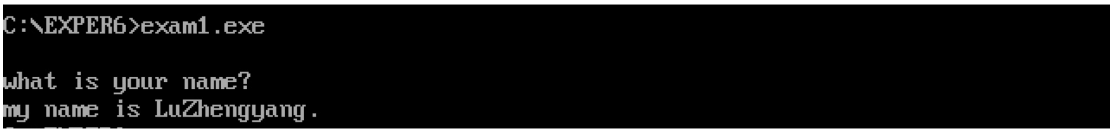
```
data segment
buf1 DB 0dh, 0ah, 'what is your name?$'
buf2 DB 0dh, 0ah, 'my name is LuZhengyang.$'
data ends

code segment
assume ds:data, cs:code

start:
   mov ax, data
   mov ds, ax

disp macro m  ; 宏汇编
   lea dx, m
   mov ah, 9
   int 21h
   endm

   disp buf1
   disp buf2
   mov ah, 4ch
   int 21h
code ends
end start
```

二. 生成可执行文件：



```
C:\EXPER6>masm exam1.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50120 + 463285 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\EXPER6>link exam1.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment
```

三. 运行结果



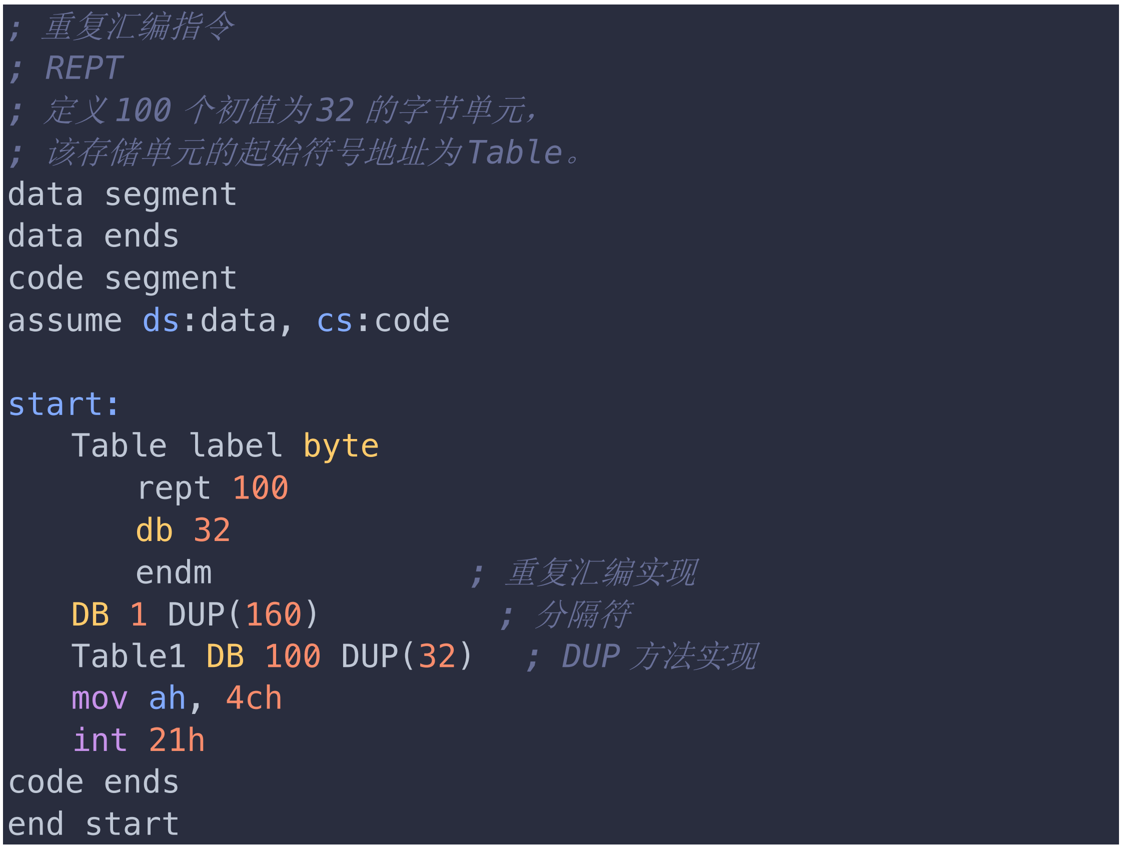
```
C:\EXPER6>exam1.exe

what is your name?
my name is LuZhengyang.
```

实验二：

一. 实现功能：

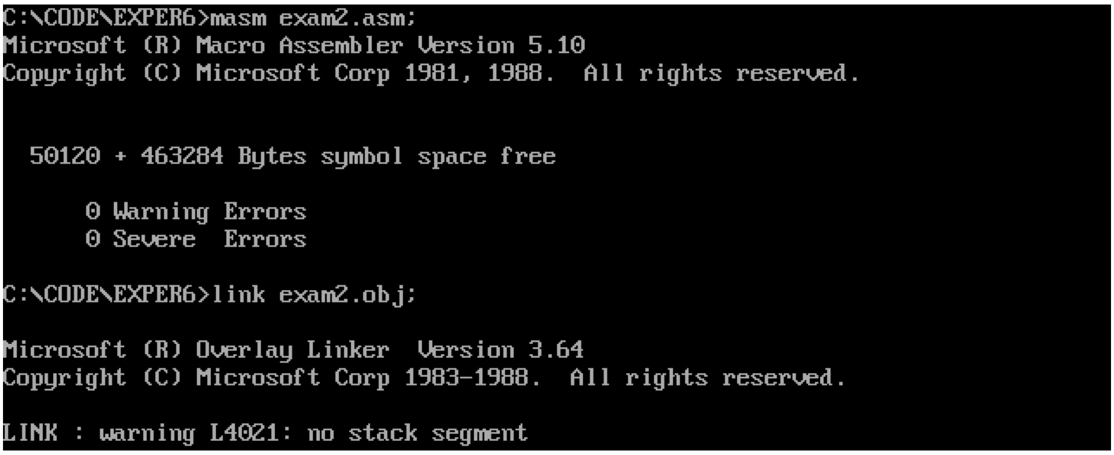
要求：定义 100 个初值为 32 的字节单元，该存储单元的起始符号地址为 Table。

代码：



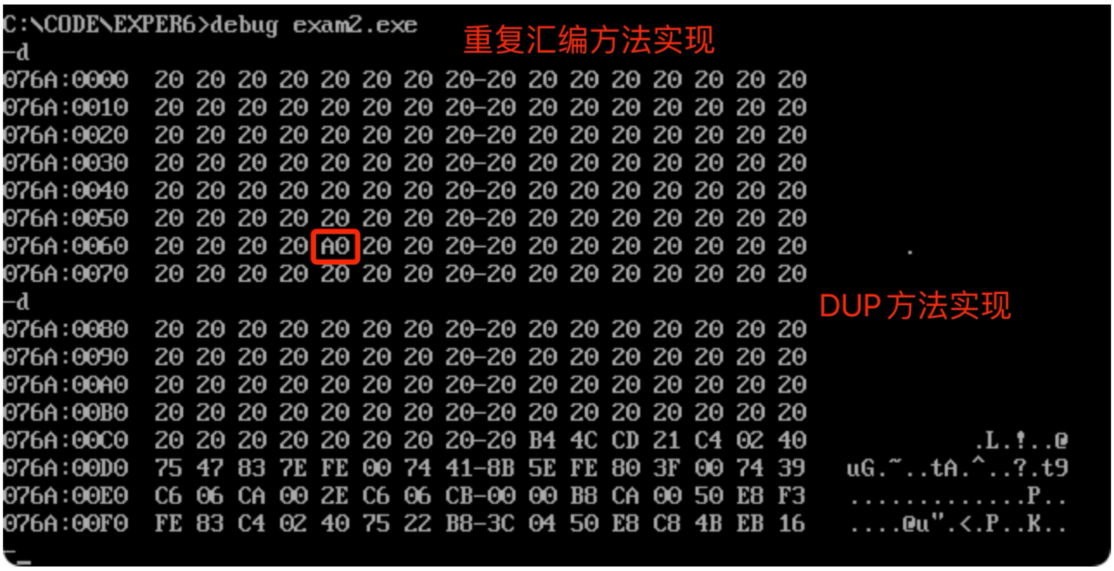
```
; 重复汇编指令
; REPT
; 定义100 个初值为32 的字节单元,
; 该存储单元的起始符号地址为Table。
data segment
data ends
code segment
assume ds:data, cs:code

start:
    Table label byte
        rept 100
        db 32
        endm                 ; 重复汇编实现
    DB 1 DUP(160)            ; 分隔符
    Table1 DB 100 DUP(32)    ; DUP 方法实现
    mov ah, 4ch
    int 21h
code ends
end start
```

二. 生成可执行文件：



```
C:\CODE\EXPER6>masm exam2.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50120 + 463284 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam2.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment
```

三. 运行结果



重复汇编方法实现

DUP 方法实现

```
C:\CODE\EXPER6>debug exam2.exe
-d
076A:0000  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0010  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0020  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0030  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0040  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0050  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0060  20 20 20 20 A0 20 20 20-20 20 20 20 20 20 20 20       .
076A:0070  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
-d
076A:0080  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:0090  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:00A0  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:00B0  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
076A:00C0  20 20 20 20 20 20 20 20-20 B4 4C CD 21 C4 02 40           .L.!..@
076A:00D0  75 47 83 7E FE 00 74 41-8B 5E FE 80 3F 00 74 39   uG.~..tA.^..?.t9
076A:00E0  C6 06 CA 00 2E C6 06 CB-00 00 B8 CA 00 50 E8 F3   .............P..
076A:00F0  FE 83 C4 02 40 75 22 B8-3C 04 50 E8 C8 4B EB 16   ....@u".<.P..K..
-
```

实验三：

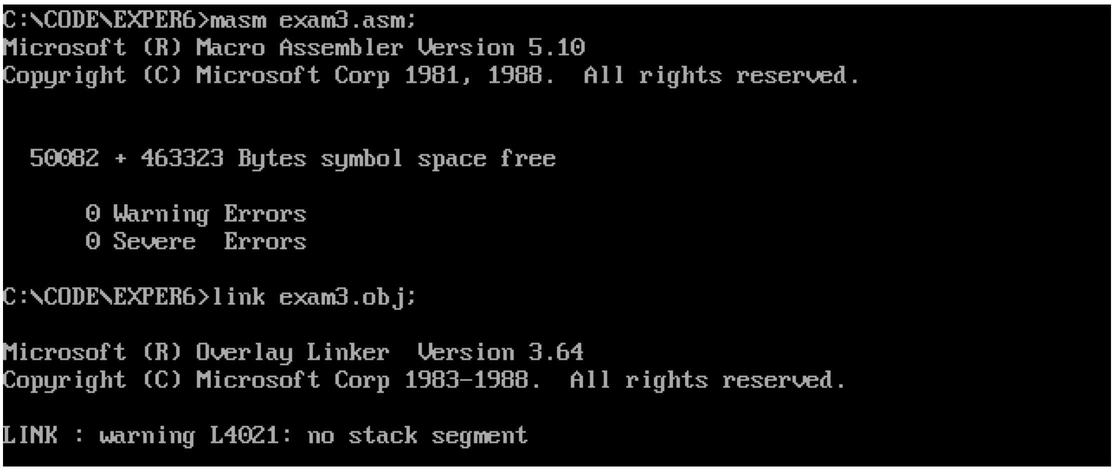
一. 实现功能：

要求：使用宏汇编功能将 A-Z 这 26 个大写字母放入数组 CHAR 中；

代码：

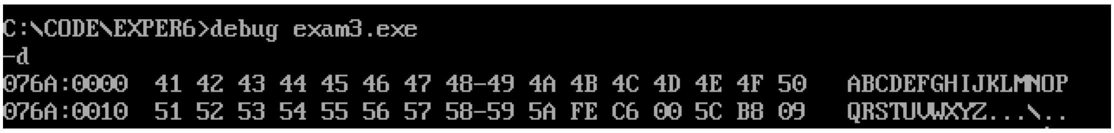
```
data segment
data ends

code segment
assume ds:data, cs:code

start:
    char label byte ; char 是标签
                    ; label 是数组关键字
                    ; byte 是数组中数据的类型
    chr = 'A'         ; 数组初始化
    rept 26           ; 重复汇编
    db chr
    chr = chr + 1
    endm              ; 重复汇编 end
code ends
end start
```

二. 生成可执行文件：



```
C:\CODE\EXPER6>masm exam3.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50082 + 463323 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam3.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment
```

三. 运行结果



```
C:\CODE\EXPER6>debug exam3.exe
-d
076A:0000  41 42 43 44 45 46 47 48-49 4A 4B 4C 4D 4E 4F 50   ABCDEFGHIJKLMNOP
076A:0010  51 52 53 54 55 56 57 58-59 5A FE C6 00 5C B8 09   QRSTUVWXYZ...\..
```

实验四：

一. 实现功能：

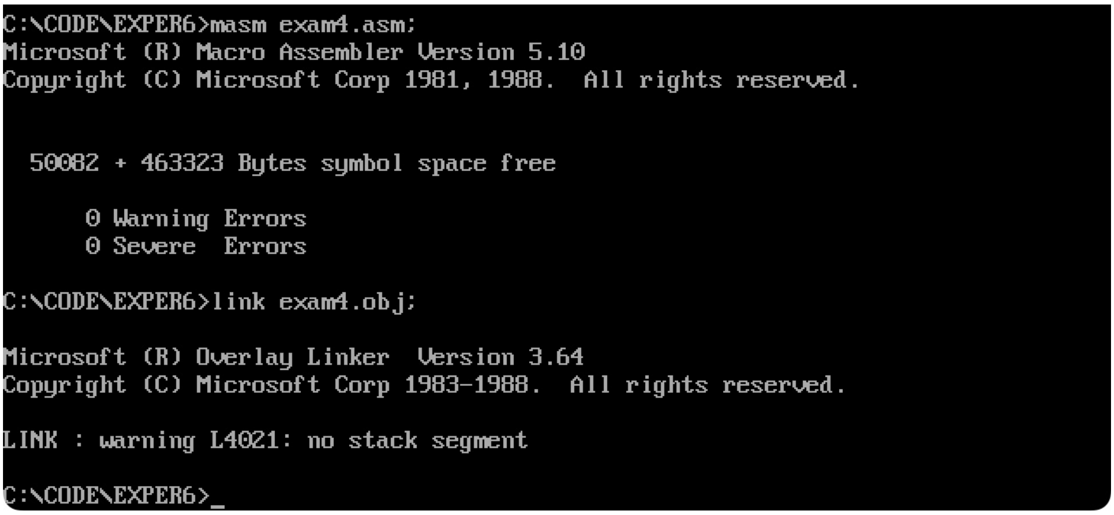
要求： 计算 1+2+ … +100，并把其值存入寄存器 AX 。

代码：

```
; 计算 1+2+ … +100，并把其值存入寄存器 AX 。
data segment
data ends

code segment
assume ds:data, cs:code
```

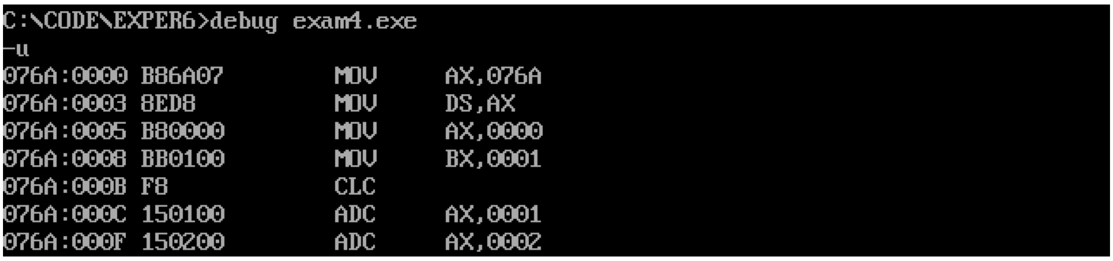
```
start:
    mov ax, data
    mov ds, ax
    ; 宏汇编实现
    mov ax, 0
    mov bx, 1
    clc
    count = 1
    rept 100
    adc ax, count
    count = count + 1
    endm
code ends
end start
```

二. 生成可执行文件：

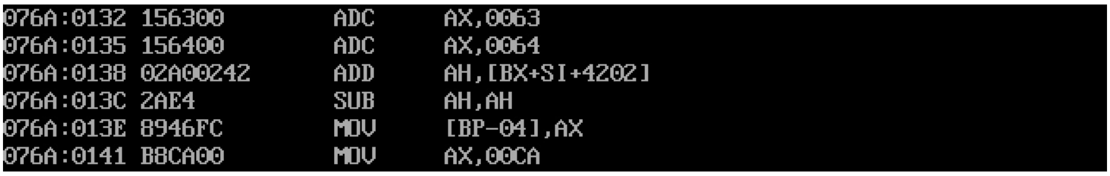


```
C:\CODE\EXPER6>masm exam4.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50082 + 463323 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam4.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment

C:\CODE\EXPER6>_
```

三. 运行结果



```
C:\CODE\EXPER6>debug exam4.exe
-u
076A:0000 B86A07        MOV     AX,076A
076A:0003 8ED8          MOV     DS,AX
076A:0005 B80000        MOV     AX,0000
076A:0008 BB0100        MOV     BX,0001
076A:000B F8            CLC
076A:000C 150100        ADC     AX,0001
076A:000F 150200        ADC     AX,0002
```



```
076A:0132 156300        ADC     AX,0063
076A:0135 156400        ADC     AX,0064
076A:0138 02A00242      ADD     AH,[BX+SI+4202]
076A:013C 2AE4          SUB     AH,AH
076A:013E 8946FC        MOV     [BP-04],AX
076A:0141 B8CA00        MOV     AX,00CA
```

实验五：

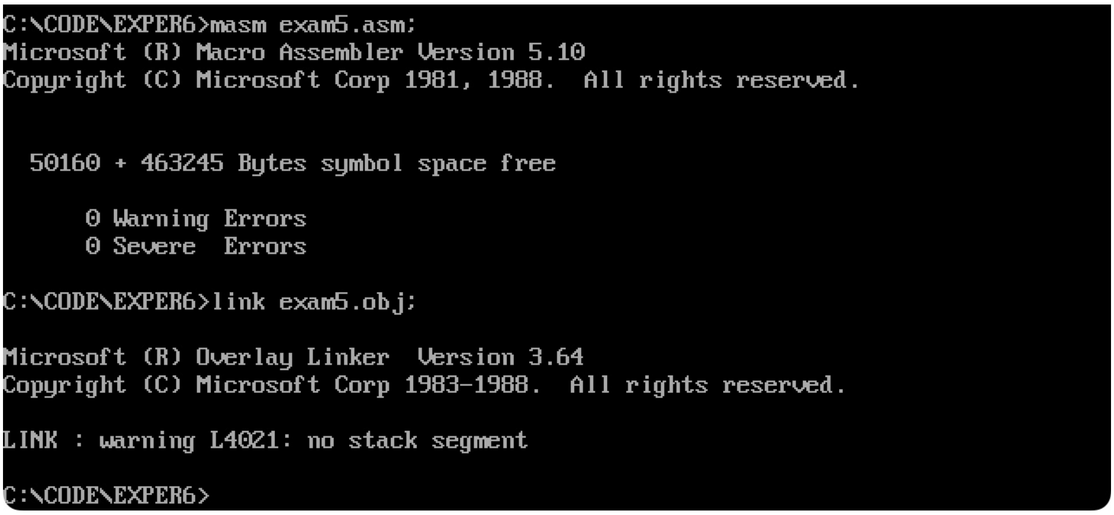
一. 实现功能：

要求：产生将 AX、BX、CX 和 DX 压 入堆栈的指令，使用 IRP 汇编指令。

代码：

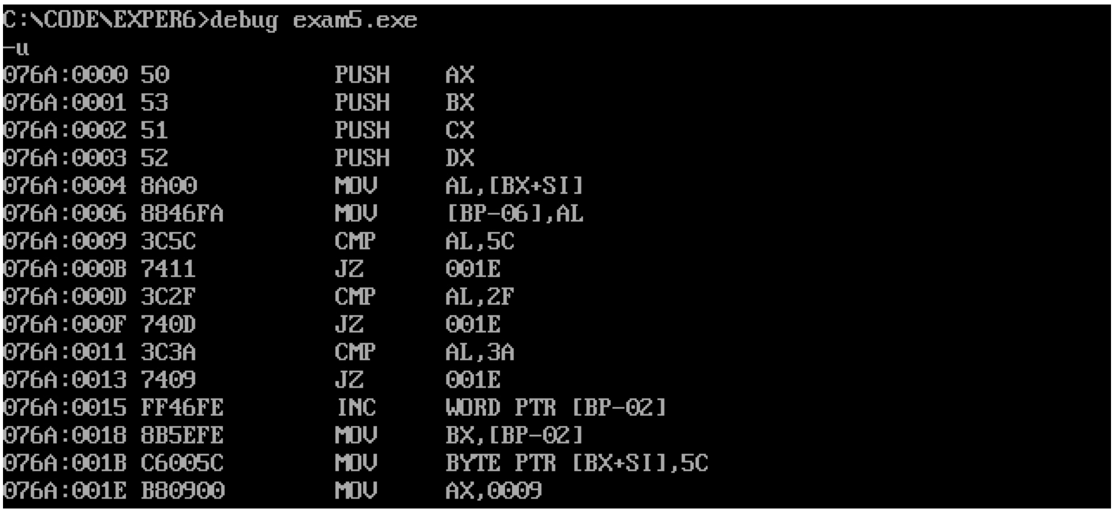
```
; 伪指令 IRP 重复次数不确定
data segment
data ends
code segment
assume ds:data, cs:code
start:
   Table label near
      IRP reg,<ax, bx, cx, dx>
      ; 重复次数就是参数的个数
      push reg
      endm
code ends
end start
```

二. 生成可执行文件：



```
C:\CODE\EXPER6>masm exam5.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50160 + 463245 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam5.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment

C:\CODE\EXPER6>
```

三. 运行结果



```
C:\CODE\EXPER6>debug exam5.exe
-u
076A:0000 50            PUSH    AX
076A:0001 53            PUSH    BX
076A:0002 51            PUSH    CX
076A:0003 52            PUSH    DX
076A:0004 8A00          MOV     AL,[BX+SI]
076A:0006 8846FA        MOV     [BP-06],AL
076A:0009 3C5C          CMP     AL,5C
076A:000B 7411          JZ      001E
076A:000D 3C2F          CMP     AL,2F
076A:000F 740D          JZ      001E
076A:0011 3C3A          CMP     AL,3A
076A:0013 7409          JZ      001E
076A:0015 FF46FE        INC     WORD PTR [BP-02]
076A:0018 8B5EFE        MOV     BX,[BP-02]
076A:001B C6005C        MOV     BYTE PTR [BX+SI],5C
076A:001E B80900        MOV     AX,0009
```

实验六：

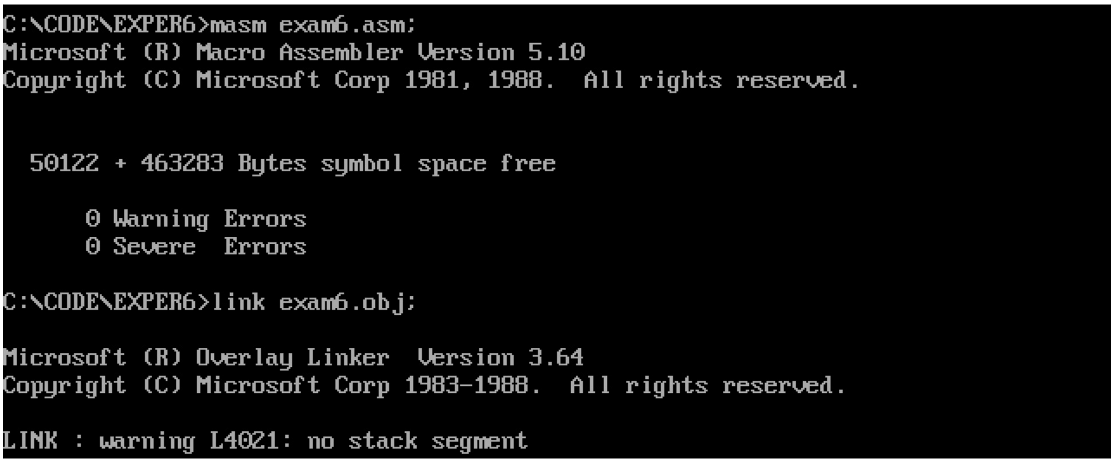
一. 实现功能：

要求：把 8 个 16 位通用寄存器之值相加，并把结果存入寄存器 AX 中，使用 IRP 宏汇编实现；

代码：

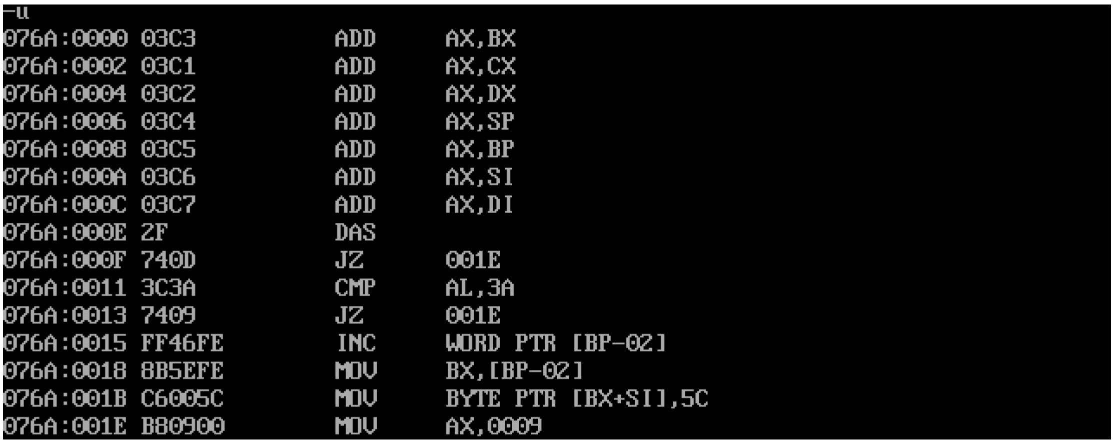
```
; 伪指令 IRP 重复次数不确定
; 把 8 个 16 位通用寄存器之值相加
; 并把结果存入寄存器 AX 中
; 由于 16 位通用寄存器名是一些不同的符号
; 不能用计数的方法来依次访问它们
; 所以我们需要用伪指令 IRP 来实现
data segment
data ends
code segment
assume ds:data, cs:code
start:
   IRP reg, <bx, cx, dx, sp, bp, si, di>
      ADD ax, reg
      endm
code ends
end start
```

二. 生成可执行文件：



```
C:\CODE\EXPER6>masm exam6.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50122 + 463283 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam6.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment
```

三. 运行结果



```
-u
076A:0000 03C3          ADD     AX,BX
076A:0002 03C1          ADD     AX,CX
076A:0004 03C2          ADD     AX,DX
076A:0006 03C4          ADD     AX,SP
076A:0008 03C5          ADD     AX,BP
076A:000A 03C6          ADD     AX,SI
076A:000C 03C7          ADD     AX,DI
076A:000E 2F            DAS
076A:000F 740D          JZ      001E
076A:0011 3C3A          CMP     AL,3A
076A:0013 7409          JZ      001E
076A:0015 FF46FE        INC     WORD PTR [BP-02]
076A:0018 8B5EFE        MOV     BX,[BP-02]
076A:001B C6005C        MOV     BYTE PTR [BX+SI],5C
076A:001E B80900        MOV     AX,0009
```

实验七：

一. 实现功能：

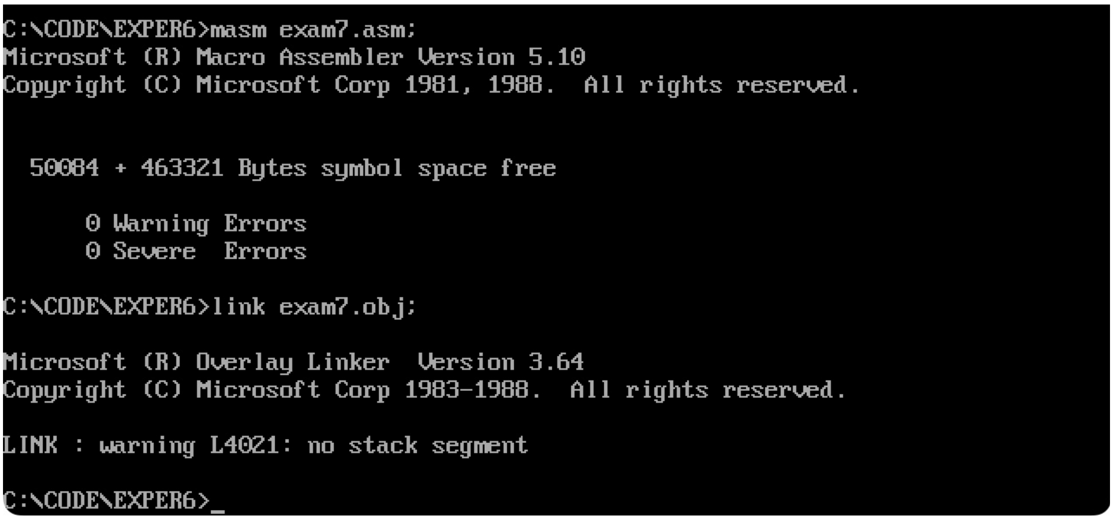
要求：定义一个元素个数不超过 100 个的数组，要求使用条件汇编；

代码：

```
; 条件汇编
; 定义一个元素个数不超过100 个的数组
data segment
data ends
code segment
assume ds:data, cs:code
start:
    PDATA MACRO NUM
        IF NUM LT 100
            DB NUM DUP(1)
```

```
       ELSE
          DB 100 DUP(2)
       ENDIF
       endm
   PDATA 99
code ends
end start
```

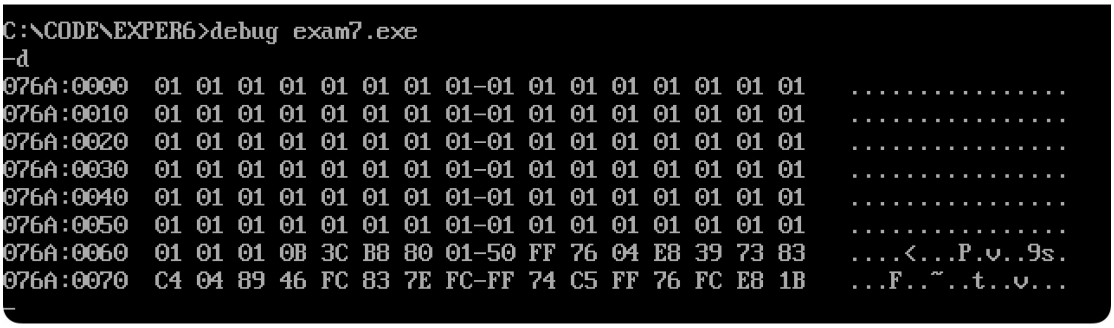
二. 生成可执行文件：



```
C:\CODE\EXPER6>masm exam7.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50084 + 463321 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link exam7.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment

C:\CODE\EXPER6>_
```

三. 运行结果

因为 99 小于 100 所以定义的数组中的元素都是 1



```
C:\CODE\EXPER6>debug exam7.exe
-d
076A:0000  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0010  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0020  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0030  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0040  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0050  01 01 01 01 01 01 01 01-01 01 01 01 01 01 01 01   ................
076A:0060  01 01 01 0B 3C B8 80 01-50 FF 76 04 E8 39 73 83   ....<...P.v..9s.
076A:0070  C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B   ...F..~..t..v...
-
```

作业：

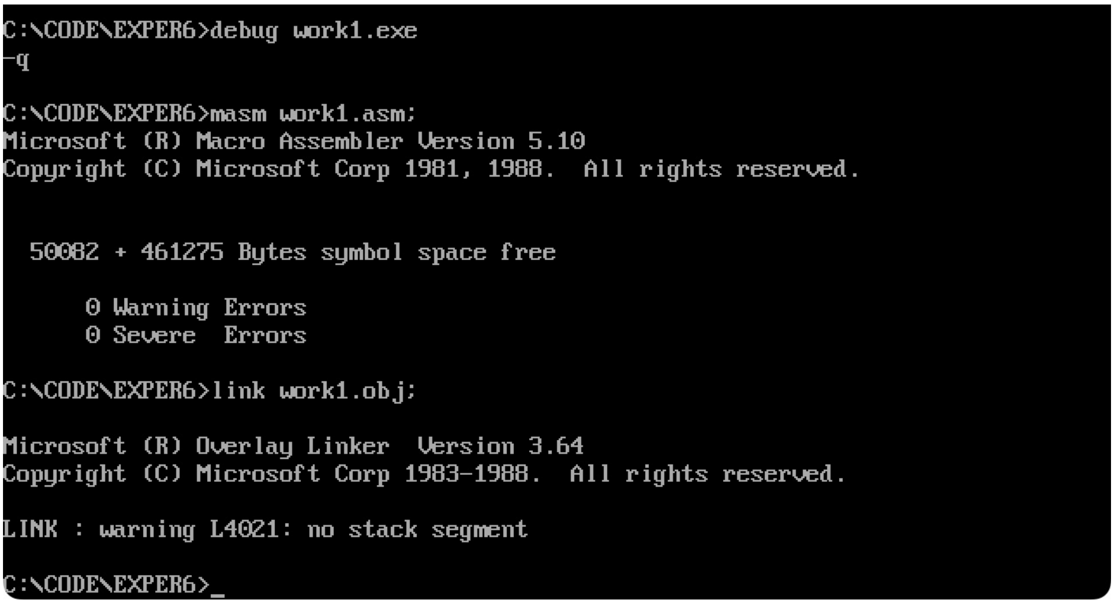
一. 实现功能：

要求：定义 100 个初值分别为 1,2,…,100 的字节单元，该存储单元的起始符号地址为 Table。

代码：

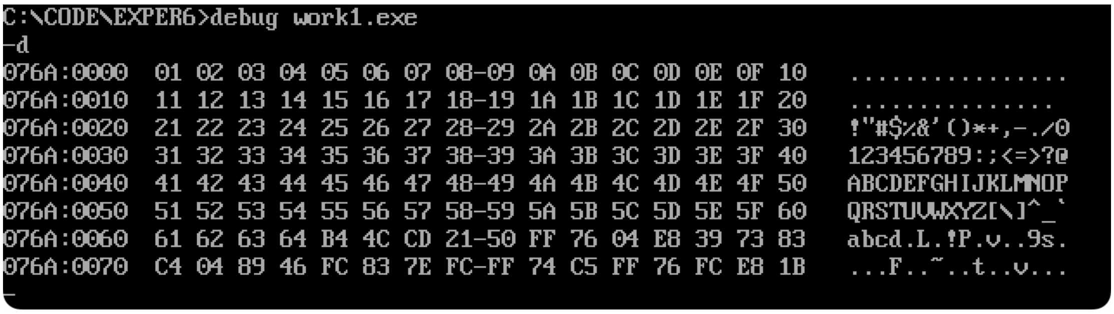
```
data segment
data ends
code segment
assume ds:data, cs:code

start:
    Table label byte
    count = 1
        rept 100
        db count
        count = count + 1
        endm            ; 重复汇编实现
    mov ah, 4ch
    int 21h
code ends
end start
```

二. 生成可执行文件：



```
C:\CODE\EXPER6>debug work1.exe
-q

C:\CODE\EXPER6>masm work1.asm;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.


  50082 + 461275 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\CODE\EXPER6>link work1.obj;

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

LINK : warning L4021: no stack segment

C:\CODE\EXPER6>_
```

三. 运行结果



```
C:\CODE\EXPER6>debug work1.exe
-d
076A:0000  01 02 03 04 05 06 07 08-09 0A 0B 0C 0D 0E 0F 10   ................
076A:0010  11 12 13 14 15 16 17 18-19 1A 1B 1C 1D 1E 1F 20   ............... 
076A:0020  21 22 23 24 25 26 27 28-29 2A 2B 2C 2D 2E 2F 30   !"#$%&'()*+,-./0
076A:0030  31 32 33 34 35 36 37 38-39 3A 3B 3C 3D 3E 3F 40   123456789:;<=>?@
076A:0040  41 42 43 44 45 46 47 48-49 4A 4B 4C 4D 4E 4F 50   ABCDEFGHIJKLMNOP
076A:0050  51 52 53 54 55 56 57 58-59 5A 5B 5C 5D 5E 5F 60   QRSTUVWXYZ[\]^_`
076A:0060  61 62 63 64 B4 4C CD 21-50 FF 76 04 E8 39 73 83   abcd.L.!P.v..9s.
076A:0070  C4 04 89 46 FC 83 7E FC-FF 74 C5 FF 76 FC E8 1B   ...F..~..t..v...
-
```

【小结或讨论】

本次实验的内容是使用宏汇编程序设计来解决一系列的问题，宏汇编实质上是将一段代码以一个标号标记，在程序中调用时宏展开，宏汇编的调用与子程序的调用是完全不同的概念，宏的调用会让程序在汇编时先进行宏展开操作，将宏对应的代码在程序中对应位置展开，展开后再进行整个程序的汇编，生成目标文件，所以宏汇编仅是源程序级的简化；宏调用在汇编时进行程序语句的展开，不需要返回；不减小目标程序，执行速度没有改变化，同时使用宏指令语句可以减少程序书写错误，缩短源程序长度，使源程序编写像高级语言一样清晰、简洁。特别是使用宏库后， 可以提高编程效率