Name : Sajid Islam   Roll No : 24p-0745
COAL

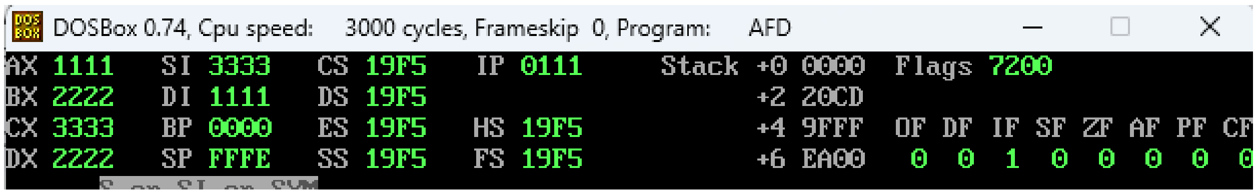
Assignment No 10

Q1

Code

```
[org 0x0100]

mov ax, 1111h
mov bx, 2222h
mov cx, 3333h

push ax
push bx
push cx

pop dx     ; DX = 3333h
pop si     ; SI = 2222h
pop di     ; DI = 1111h

xchg dx, si ; Swap DX and SI

mov ax, 0x4c00
int 0x21
```

OUTPUT

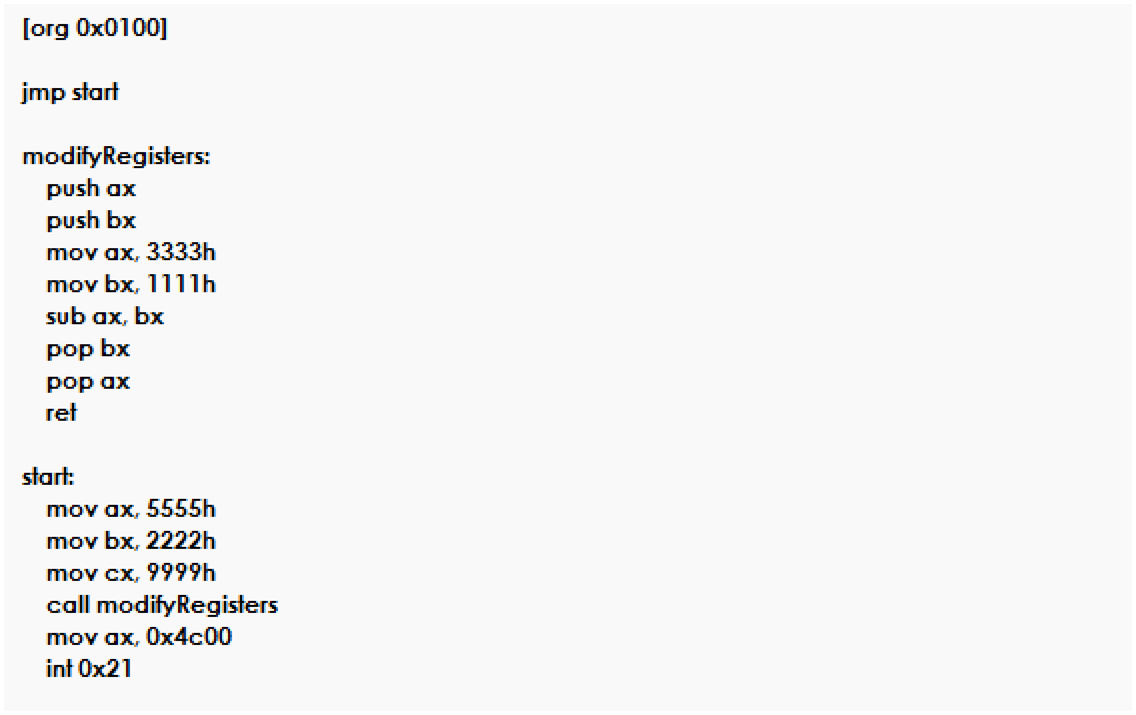


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

```
AX 1111   SI 3333   CS 19F5   IP 0111      Stack +0 0000  Flags 7200
BX 2222   DI 1111   DS 19F5                      +2 20CD
CX 3333   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 2222   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0
```

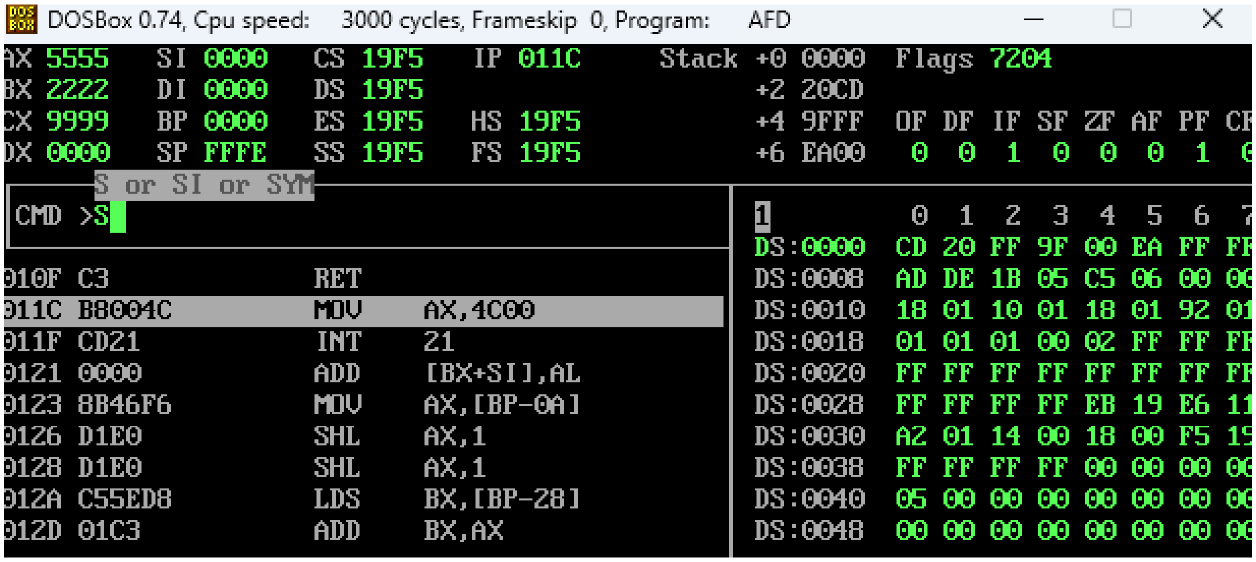
Q2

Code



```
[org 0x0100]

jmp start

modifyRegisters:
   push ax
   push bx
   mov ax, 3333h
   mov bx, 1111h
   sub ax, bx
   pop bx
   pop ax
   ret

start:
   mov ax, 5555h
   mov bx, 2222h
   mov cx, 9999h
   call modifyRegisters
   mov ax, 0x4c00
   int 0x21
```

OUTPUT

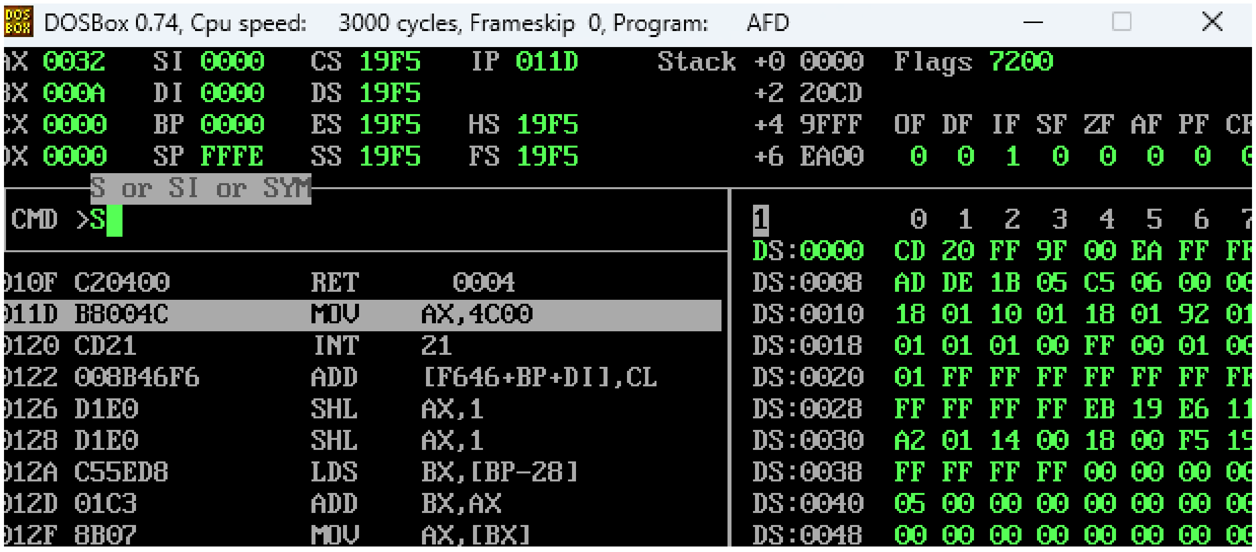

```
DOSBox 0.74, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 5555   SI 0000   CS 19F5   IP 011C      Stack +0 0000  Flags 7204
BX 2222   DI 0000   DS 19F5                      +2 20CD
CX 9999   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  1  0
            S or SI or SYM
 CMD >S                                          1          0  1  2  3  4  5  6  7
                                                 DS:0000  CD 20 FF 9F 00 EA FF FF
010F C3             RET                          DS:0008  AD DE 1B 05 C5 06 00 00
011C B8004C         MOV     AX,4C00              DS:0010  18 01 10 01 18 01 92 01
011F CD21           INT     21                   DS:0018  01 01 01 00 02 FF FF FF
0121 0000           ADD     [BX+SI],AL           DS:0020  FF FF FF FF FF FF FF FF
0123 8B46F6         MOV     AX,[BP-0A]           DS:0028  FF FF FF FF EB 19 E6 11
0126 D1E0           SHL     AX,1                 DS:0030  A2 01 14 00 18 00 F5 19
0128 D1E0           SHL     AX,1                 DS:0038  FF FF FF FF 00 00 00 00
012A C55ED8         LDS     BX,[BP-28]           DS:0040  05 00 00 00 00 00 00 00
012D 01C3           ADD     BX,AX                DS:0048  00 00 00 00 00 00 00 00
```

Q3

Code

```
[org 0x0100]

jmp start

multiply:
   push bp
   mov bp, sp
   mov ax, [bp+6]
   mov bx, [bp+4]
   mul bx
   pop bp
   ret 4

start:
   mov ax, 5
   push ax
   mov ax, 10
   push ax
   call multiply
   mov ax, 0x4c00
   int 0x21
```

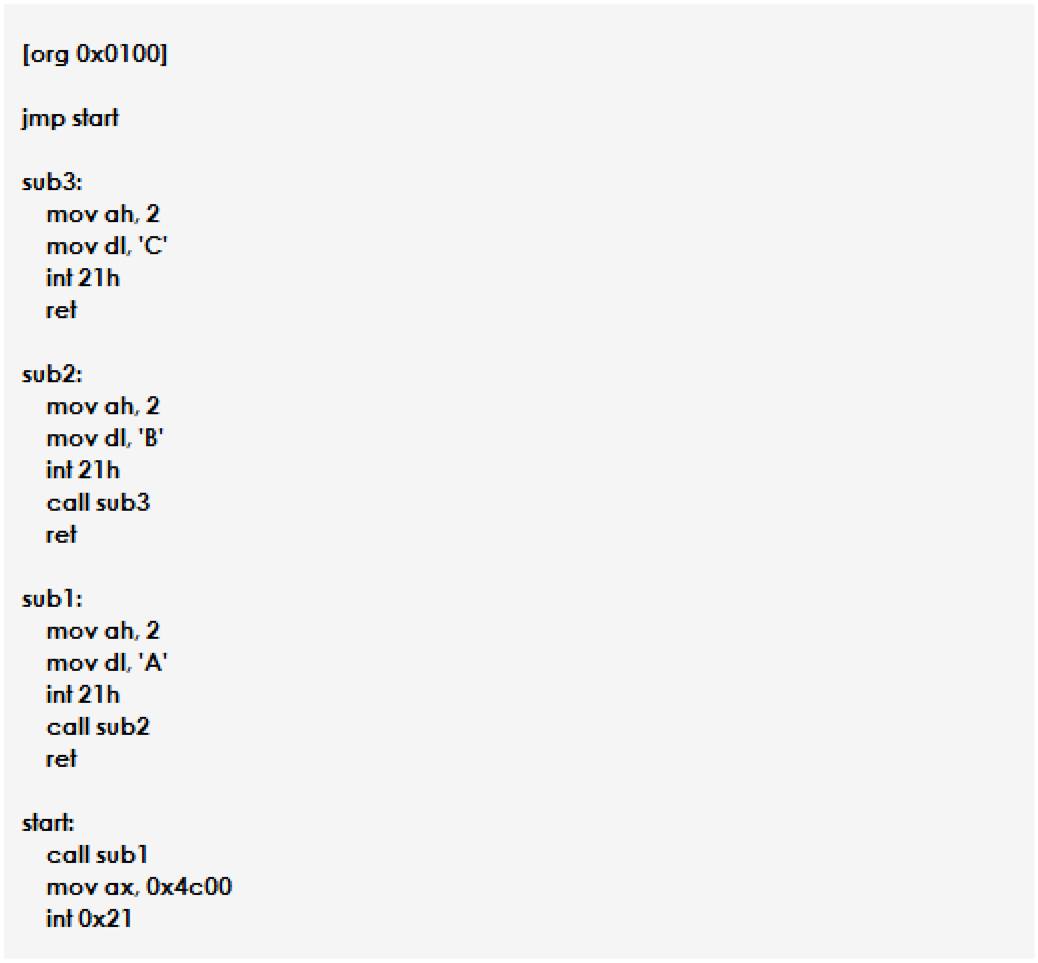
OUTPUT



```
DOSBox 0.74, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0032   SI 0000   CS 19F5    IP 011D      Stack +0 0000  Flags 7200
BX 000A   DI 0000   DS 19F5                       +2 20CD
CX 0000   BP 0000   ES 19F5    HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5    FS 19F5            +6 EA00   0  0  1  0  0  0  0  0
     S or SI or SYM
CMD >S                                         1          0  1  2  3  4  5  6  7
                                               DS:0000   CD 20 FF 9F 00 EA FF FF
010F C20400          RET      0004             DS:0008   AD DE 1B 05 C5 06 00 00
011D B8004C          MOV    AX,4C00            DS:0010   18 01 10 01 18 01 92 01
0120 CD21            INT    21                 DS:0018   01 01 01 00 FF 00 01 00
0122 008B46F6        ADD    [F646+BP+DI],CL    DS:0020   01 FF FF FF FF FF FF FF
0126 D1E0            SHL    AX,1               DS:0028   FF FF FF FF EB 19 E6 11
0128 D1E0            SHL    AX,1               DS:0030   A2 01 14 00 18 00 F5 15
012A C55ED8          LDS    BX,[BP-28]         DS:0038   FF FF FF FF 00 00 00 00
012D 01C3            ADD    BX,AX              DS:0040   05 00 00 00 00 00 00 00
012F 8B07            MOV    AX,[BX]            DS:0048   00 00 00 00 00 00 00 00
```

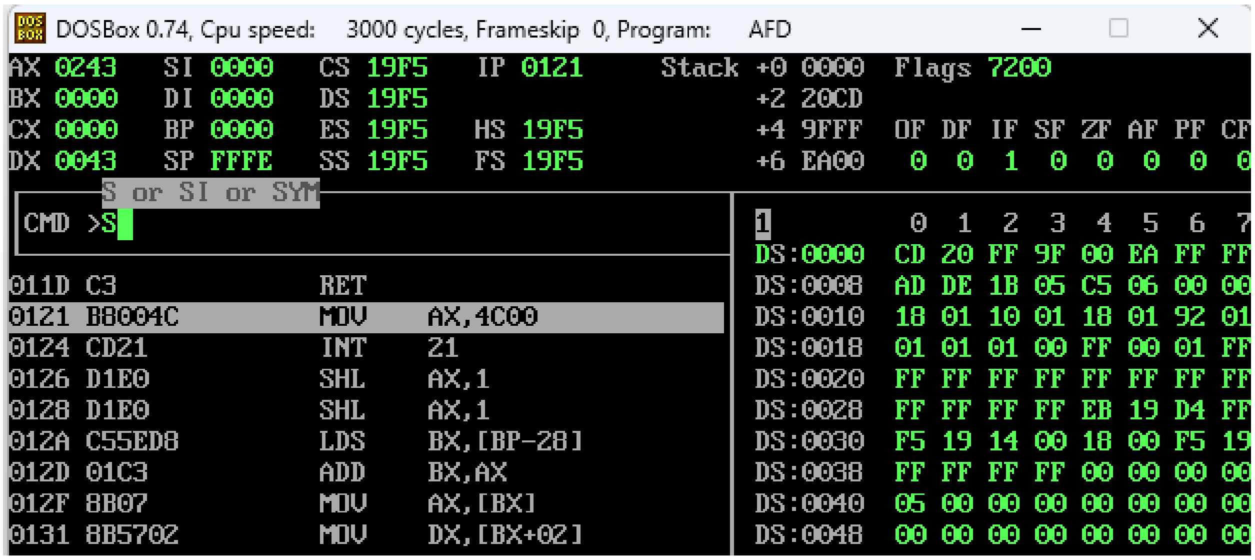
Q4

Code



```
[org 0x0100]

jmp start

sub3:
   mov ah, 2
   mov dl, 'C'
   int 21h
   ret

sub2:
   mov ah, 2
   mov dl, 'B'
   int 21h
   call sub3
   ret

sub1:
   mov ah, 2
   mov dl, 'A'
   int 21h
   call sub2
   ret

start:
   call sub1
   mov ax, 0x4c00
   int 0x21
```

OUTPUT



```
DOSBox 0.74, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0243    SI 0000    CS 19F5    IP 0121       Stack +0 0000  Flags 7200
BX 0000    DI 0000    DS 19F5                        +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5             +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0043    SP FFFE    SS 19F5    FS 19F5             +6 EA00   0  0  1  0  0  0  0  0
             S or SI or SYM
 CMD >S                                            1          0  1  2  3  4  5  6  7
                                                   DS:0000  CD 20 FF 9F 00 EA FF FF
0110 C3                RET                         DS:0008  AD DE 1B 05 C5 06 00 00
0121 B8004C            MOV     AX,4C00             DS:0010  18 01 10 01 18 01 92 01
0124 CD21              INT     21                  DS:0018  01 01 01 00 FF 00 01 FF
0126 D1E0              SHL     AX,1                DS:0020  FF FF FF FF FF FF FF FF
0128 D1E0              SHL     AX,1                DS:0028  FF FF FF FF EB 19 D4 FF
012A C55ED8            LDS     BX,[BP-28]          DS:0030  F5 19 14 00 18 00 F5 19
012D 01C3              ADD     BX,AX               DS:0038  FF FF FF FF 00 00 00 00
012F 8B07              MOV     AX,[BX]             DS:0040  05 00 00 00 00 00 00 00
0131 8B5702            MOV     DX,[BX+02]          DS:0048  00 00 00 00 00 00 00 00
```

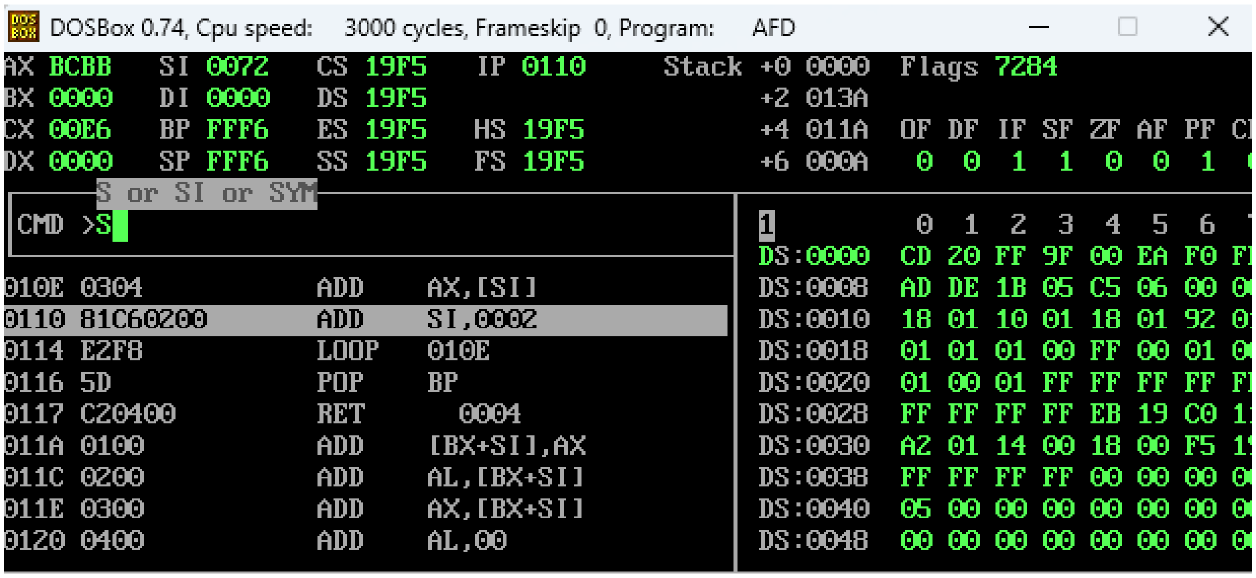
Q5

Code

```
org 0x0100

jmp start

sum_array:
   push bp
   mov  bp, sp
   mov  cx, [bp+4]     ; count
   mov  si, [bp+6]     ; address of array
   xor  ax, ax         ; sum = 0
next:
   add  ax, [si]       ; add word at [si]
   add  si, 2          ; next element
   loop next
   pop  bp
   ret  4

numbers dw 1,2,3,4,5,6,7,8,9,10

start:
   mov  ax, 10
   push ax             ; count
   lea  ax, [numbers]
   push ax             ; address
   call sum_array

   ; ax now holds the sum (55)
   mov  ax, 0x4C00
   int  0x21
```

Output

Screen shot when loop runs some time



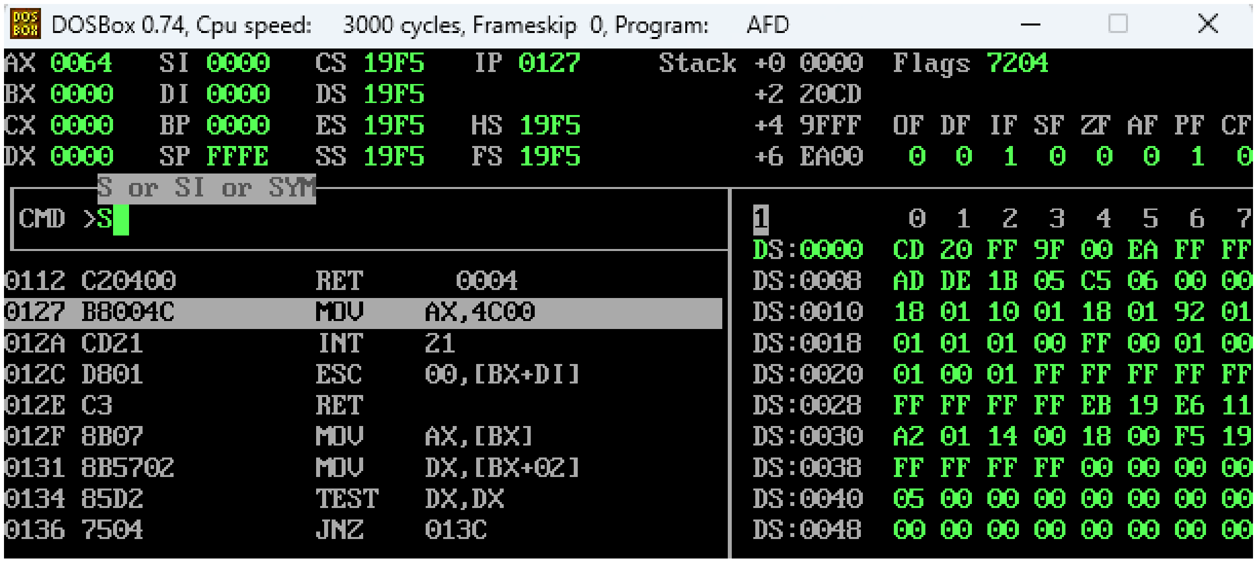
```
DOSBox 0.74, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX BCBB   SI 0072   CS 19F5   IP 0110     Stack +0 0000  Flags 7284
BX 0000   DI 0000   DS 19F5                     +2 013A
CX 00E6   BP FFF6   ES 19F5   HS 19F5           +4 011A  OF DF IF SF ZF AF PF C
DX 0000   SP FFF6   SS 19F5   FS 19F5           +6 000A   0  0  1  1  0  0  1
      S or SI or SYM
 CMD >S                                        1          0  1  2  3  4  5  6
                                               DS:0000  CD 20 FF 9F 00 EA F0 F
010E 0304          ADD     AX,[SI]             DS:0008  AD DE 1B 05 C5 06 00 0
0110 81C60200      ADD     SI,0002             DS:0010  18 01 10 01 18 01 92 0
0114 E2F8          LOOP    010E                DS:0018  01 01 01 00 FF 00 01 0
0116 5D            POP     BP                  DS:0020  01 00 01 FF FF FF FF F
0117 C20400        RET       0004              DS:0028  FF FF FF FF EB 19 C0 1
011A 0100          ADD     [BX+SI],AX          DS:0030  A2 01 14 00 18 00 F5 1
011C 0200          ADD     AL,[BX+SI]          DS:0038  FF FF FF FF 00 00 00 0
011E 0300          ADD     AX,[BX+SI]          DS:0040  05 00 00 00 00 00 00 0
0120 0400          ADD     AL,00               DS:0048  00 00 00 00 00 00 00 0
```

Q6

Code

```
[org 0x0100]
jmp start


; [bp+6] holds the first parameter
; [bp+4] holds the second parameter
findMin:
   push bp           ; 1. Save old BP
   mov bp, sp         ; 2. Set up stack frame

   mov ax, [bp+6]
   cmp ax, [bp+4]
   jle is_smaller     ; If AX <= [bp+4], AX is already the min

   mov ax, [bp+4]     ; Otherwise, [bp+4] was smaller, move it to AX

is_smaller:
   pop bp            ; 5. Restore old BP
   ret 4


start:
   ; Test 1: (900, 1200) -> AX should be 900
   push 900
   push 1200
   call findMin

   push 500
   push 100
   call findMin

   mov ax, 0x4c00     ; Terminate program
   int 0x21
```

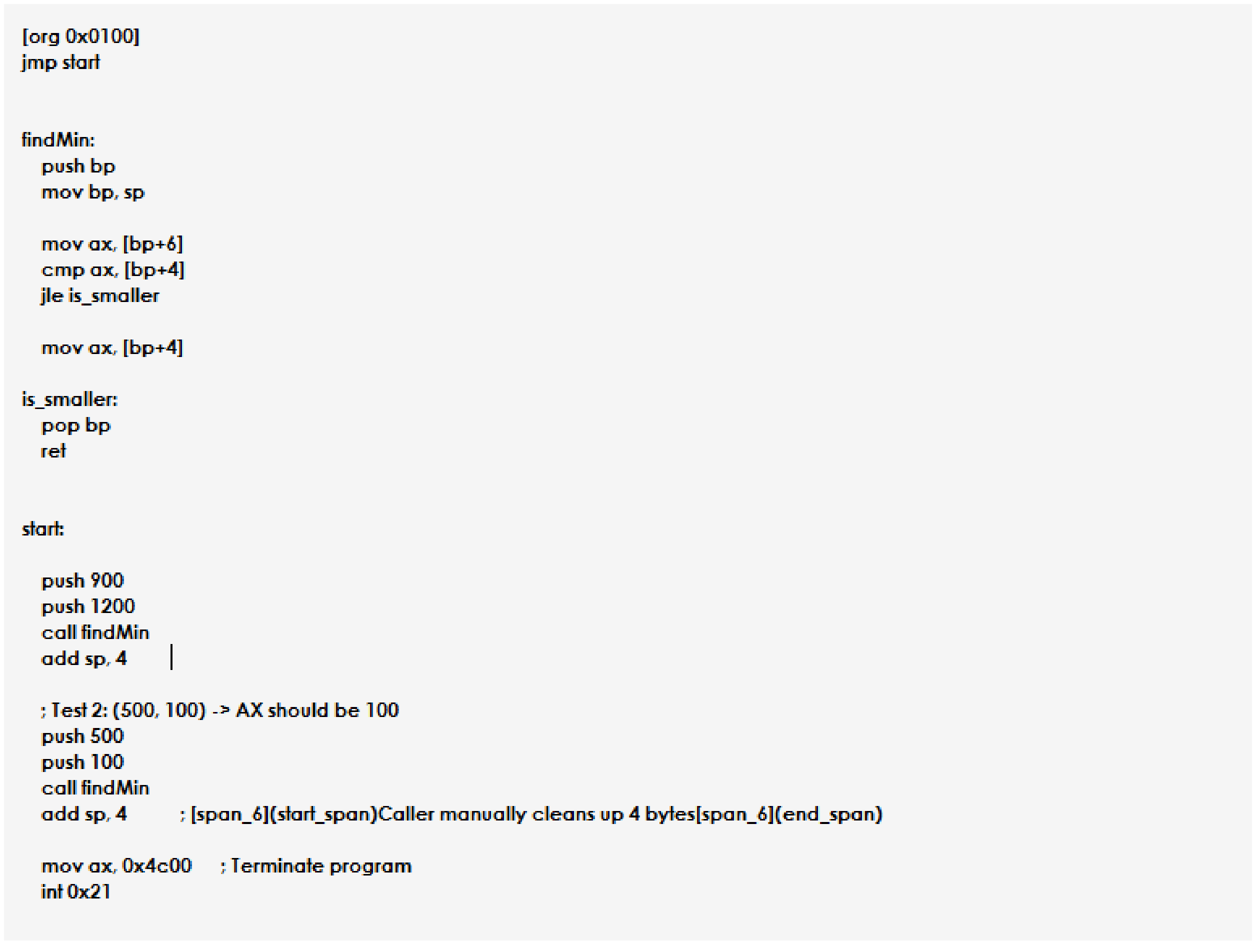
Output


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

```
AX 0064    SI 0000    CS 19F5    IP 0127      Stack +0 0000  Flags 7204
BX 0000    DI 0000    DS 19F5                       +2 20CD
CX 0000    BP 0000    ES 19F5    HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000    SP FFFE    SS 19F5    FS 19F5            +6 EA00   0  0  1  0  0  0  1  0
      S or SI or SYM
 CMD >S                                          1          0  1  2  3  4  5  6  7
                                                 DS:0000  CD 20 FF 9F 00 EA FF FF
0112 C20400         RET      0004                DS:0008  AD DE 1B 05 C5 06 00 00
0127 B8004C         MOV     AX,4C00              DS:0010  18 01 10 01 18 01 92 01
012A CD21           INT     21                   DS:0018  01 01 01 00 FF 00 01 00
012C D801           ESC     00,[BX+DI]           DS:0020  01 00 01 FF FF FF FF FF
012E C3             RET                          DS:0028  FF FF FF FF EB 19 E6 11
012F 8B07           MOV     AX,[BX]              DS:0030  A2 01 14 00 18 00 F5 19
0131 8B5702         MOV     DX,[BX+02]           DS:0038  FF FF FF FF 00 00 00 00
0134 85D2           TEST    DX,DX                DS:0040  05 00 00 00 00 00 00 00
0136 7504           JNZ     013C                 DS:0048  00 00 00 00 00 00 00 00
```

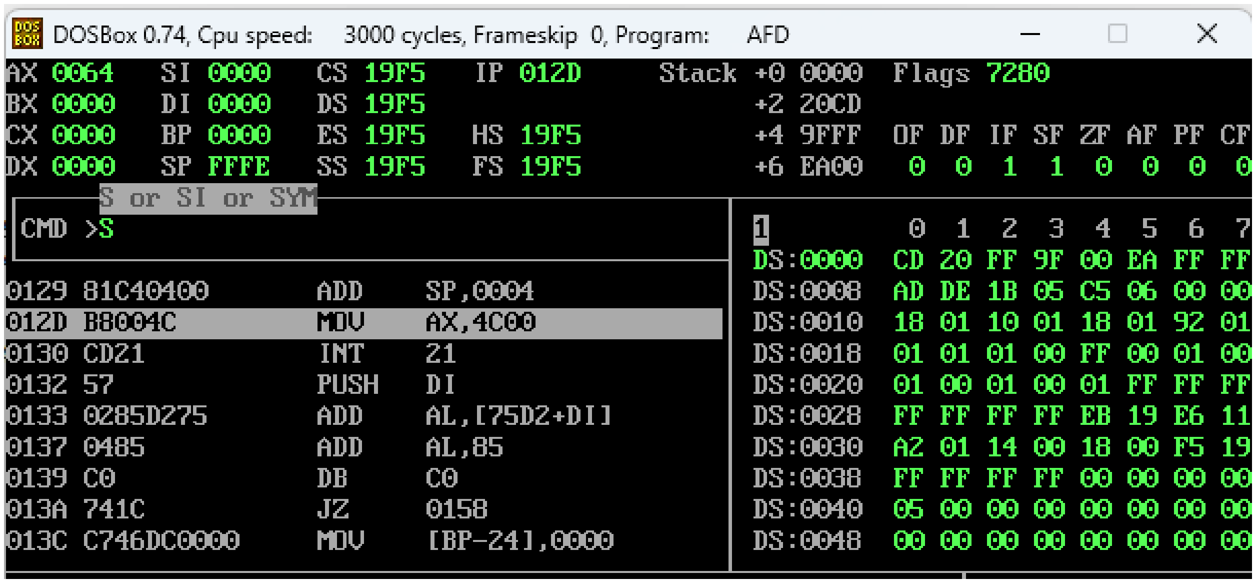
Q7

Code



```
[org 0x0100]
jmp start


findMin:
   push bp
   mov bp, sp

   mov ax, [bp+6]
   cmp ax, [bp+4]
   jle is_smaller

   mov ax, [bp+4]

is_smaller:
   pop bp
   ret


start:

   push 900
   push 1200
   call findMin
   add sp, 4

   ; Test 2: (500, 100) -> AX should be 100
   push 500
   push 100
   call findMin
   add sp, 4        ; [span_6](start_span)Caller manually cleans up 4 bytes[span_6](end_span)

   mov ax, 0x4c00     ; Terminate program
   int 0x21
```

Output

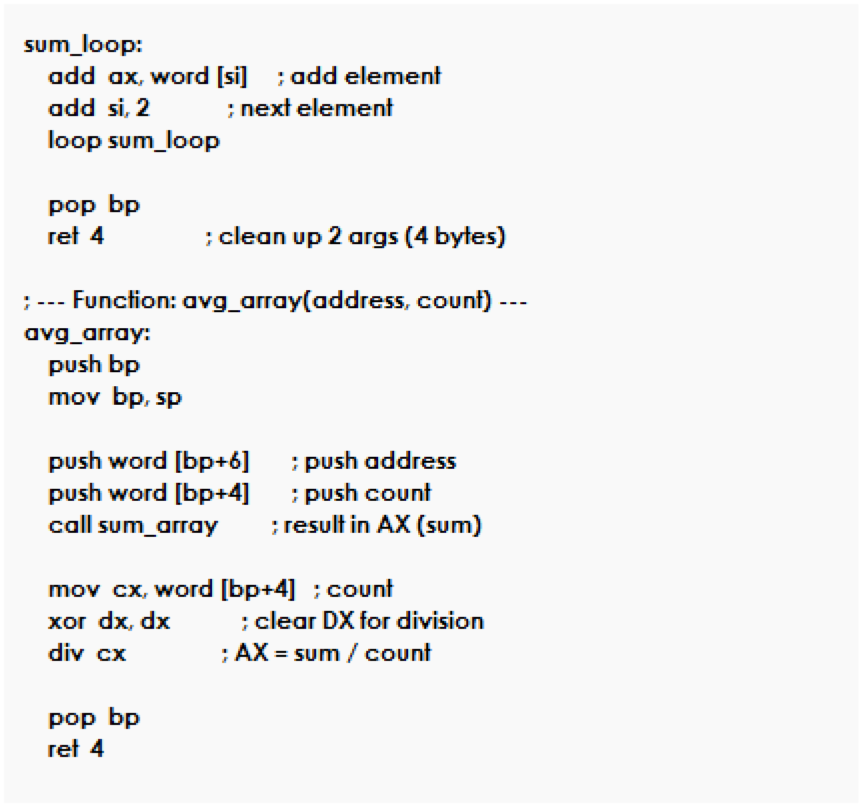


```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   AFD
AX 0064   SI 0000   CS 19F5   IP 012D     Stack +0 0000  Flags 7280
BX 0000   DI 0000   DS 19F5                     +2 20CD
CX 0000   BP 0000   ES 19F5   HS 19F5           +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5           +6 EA00   0  0  1  1  0  0  0  0
    S or SI or SYM
 CMD >S                                         1         0  1  2  3  4  5  6  7
                                                DS:0000  CD 20 FF 9F 00 EA FF FF
0129 81C40400       ADD    SP,0004              DS:0008  AD DE 1B 05 C5 06 00 00
012D B8004C         MOV    AX,4C00              DS:0010  18 01 10 01 18 01 92 01
0130 CD21           INT    21                   DS:0018  01 01 01 00 FF 00 01 00
0132 57             PUSH   DI                   DS:0020  01 00 01 00 01 FF FF FF
0133 0285D275       ADD    AL,[75D2+DI]         DS:0028  FF FF FF FF EB 19 E6 11
0137 0485           ADD    AL,85                DS:0030  A2 01 14 00 18 00 F5 19
0139 C0             DB     C0                   DS:0038  FF FF FF FF 00 00 00 00
013A 741C           JZ     0158                 DS:0040  05 00 00 00 00 00 00 00
013C C746DC0000     MOV    [BP-24],0000         DS:0048  00 00 00 00 00 00 00 00
```

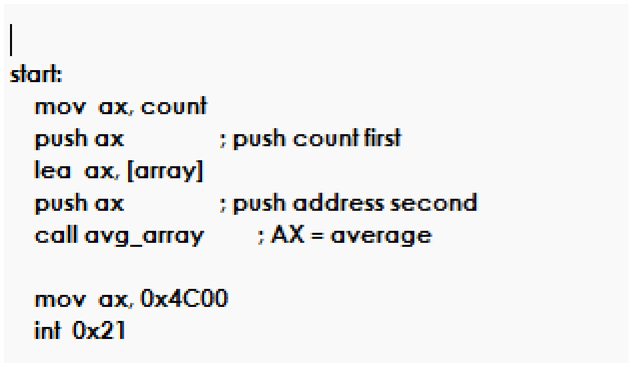
Q8

Code

```
org 0x0100
jmp start


array dw 10, 20, 30, 40, 50
count equ 5

sum_array:
   push bp
   mov  bp, sp

   mov  si, word [bp+6]   ; starting address of array
   mov  cx, word [bp+4]   ; count
   xor  ax, ax           ; sum = 0
```



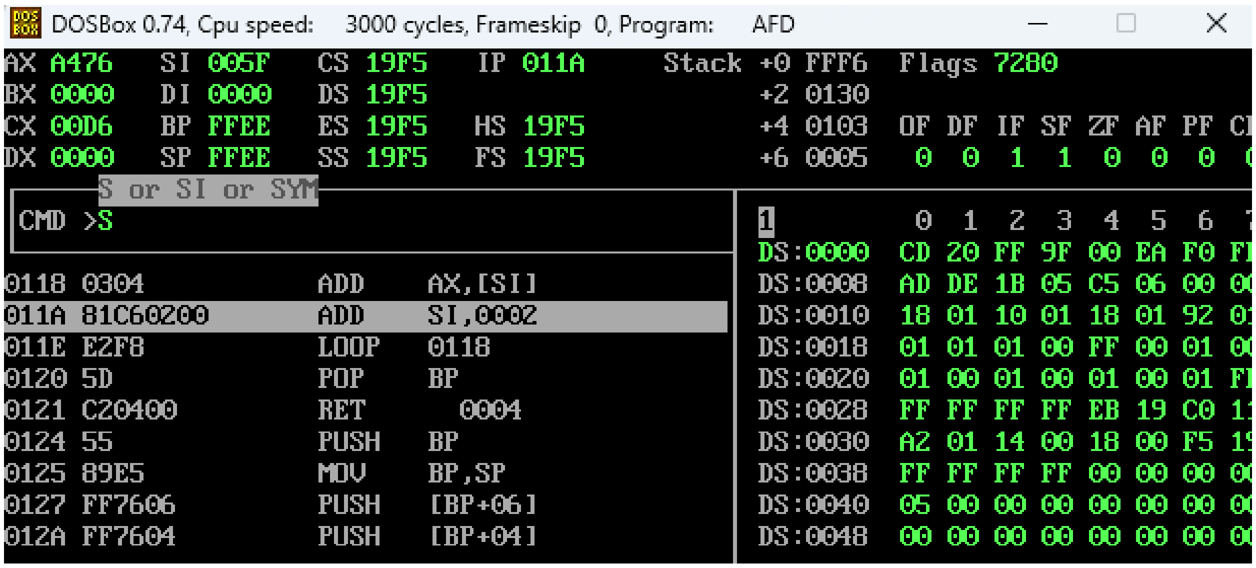
```
sum_loop:
   add  ax, word [si]    ; add element
   add  si, 2           ; next element
   loop sum_loop

   pop  bp
   ret  4              ; clean up 2 args (4 bytes)

; --- Function: avg_array(address, count) ---
avg_array:
   push bp
   mov  bp, sp

   push word [bp+6]      ; push address
   push word [bp+4]      ; push count
   call sum_array        ; result in AX (sum)

   mov  cx, word [bp+4]   ; count
   xor  dx, dx          ; clear DX for division
   div  cx              ; AX = sum / count

   pop  bp
   ret  4
```



```
start:
   mov  ax, count
   push ax              ; push count first
   lea  ax, [array]
   push ax              ; push address second
   call avg_array       ; AX = average

   mov  ax, 0x4C00
   int  0x21
```

Output

Screen after multiple loop iteration happens



```
DOSBox 0.74, Cpu speed:     3000 cycles, Frameskip  0, Program:     AFD
AX A476   SI 005F   CS 19F5   IP 011A     Stack +0 FFF6  Flags 7280
BX 0000   DI 0000   DS 19F5                     +2 0130
CX 00D6   BP FFEE   ES 19F5   HS 19F5           +4 0103  OF DF IF SF ZF AF PF CI
DX 0000   SP FFEE   SS 19F5   FS 19F5           +6 0005   0  0  1  1  0  0  0  0
            S or SI or SYM
 CMD >S                                            1         0  1  2  3  4  5  6  7
                                                   DS:0000  CD 20 FF 9F 00 EA F0 FI
0118 0304               ADD     AX,[SI]            DS:0008  AD DE 1B 05 C5 06 00 00
011A 81C60200           ADD     SI,0002            DS:0010  18 01 10 01 18 01 92 01
011E E2F8               LOOP    0118               DS:0018  01 01 01 00 FF 00 01 00
0120 5D                 POP     BP                 DS:0020  01 00 01 00 01 00 01 FI
0121 C20400             RET       0004             DS:0028  FF FF FF FF EB 19 C0 11
0124 55                 PUSH    BP                 DS:0030  A2 01 14 00 18 00 F5 19
0125 89E5               MOV     BP,SP              DS:0038  FF FF FF FF 00 00 00 00
0127 FF7606             PUSH    [BP+06]            DS:0040  05 00 00 00 00 00 00 00
012A FF7604             PUSH    [BP+04]            DS:0048  00 00 00 00 00 00 00 00
```