Mawlana Bhashani Science and Technology University

Department of Information and Communication Technology

Lab Report

Course Title: Microprocessor and Embedded System Lab

Course Code: ICT-2204

Index

Submitted by	Submitted to
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	Dr. Abir Hossain Associate Professor, Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Course Title: Microprocessor And Embedded System Lab

Course Code: ICT-2204

Experiment name: Installation of EMU8086 Microprocessor Emulator.

Lab Report No: 01

Submitted By	Submitted To
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	**Dr. Md. Abir Hossain** Associate Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

Date of Performance: 13/11/2024
Date of Submission: 20/11/2024

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Course Title: Microprocessor And Embedded System Lab

Course Code: ICT-2204

Lab Report No: 02

Submitted By	Submitted To
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	**Dr. Md. Abir Hossain** Associate Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

Date of Performance: 20/11/2024
Date of Submission: 27/11/2024

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Course Title: Microprocessor And Assembly Language.

Course Code: ICT-2204

Lab Report No: 03

Submitted By	Submitted To
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	**Dr. Md. Abir Hossain** Associate Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

Date of Performance: 27 /11/2024

Date of Submission: 04/12/2024

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Course Title: Microprocessor And Embedded System Lab

Course Code: ICT-2204

Lab Report No: 04

Submitted By	Submitted To
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	**Dr. Md. Abir Hossain** Associate Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

Date of Performance: 04/12/2024
Date of Submission: 11/12/2024

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Course Title: Microprocessor And Embedded System Lab

Course Code: ICT-2204

Lab Report No: 05

Submitted By	Submitted To
Name: Afifa Tahsin Haq ID: IT22005 2nd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	**Dr. Md. Abir Hossain** Associate Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

Date of Performance: 04/12/2024
Date of Submission: 11/12/2024

Experiment no: 01

Experiment name: Installation of EMU8086 Microprocessor Emulator

Introduction: EMU8086 is a comprehensive microprocessor emulator that enables users to write, compile, and debug assembly code for the 8086 microprocessor. It is commonly used in academic settings for teaching assembly language programming concepts and microprocessor architecture.

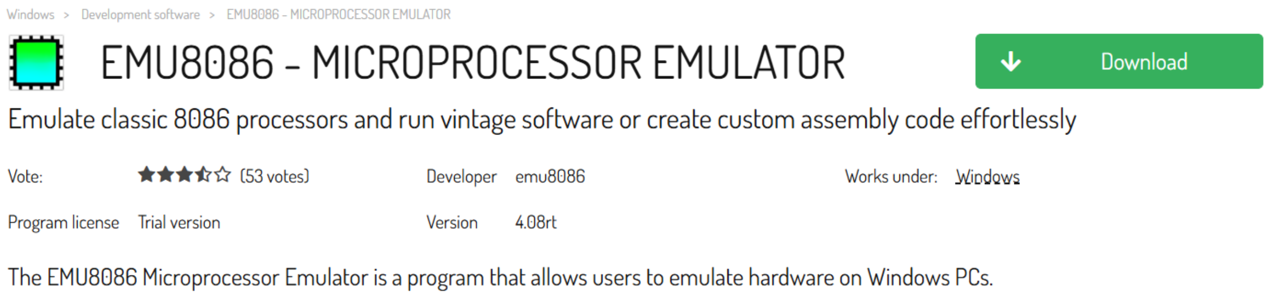
Objectives: To successfully download, install, and set up EMU8086, a microprocessor emulator used for learning and simulating Assembly language programming.

Required Instruments:

1. A computer with Windows operating system.
2. Internet connection.
3. EMU8086 installation package (can be downloaded from the official website or trusted sources).

Procedure:

Step 1: Download EMU8086: Open a web browser and navigate to the official EMU8086 website or another trusted source. Locate the latest version of EMU8086 and download the setup file. Click on Download button.


Windows > Development software > EMU8086 - MICROPROCESSOR EMULATOR
Download

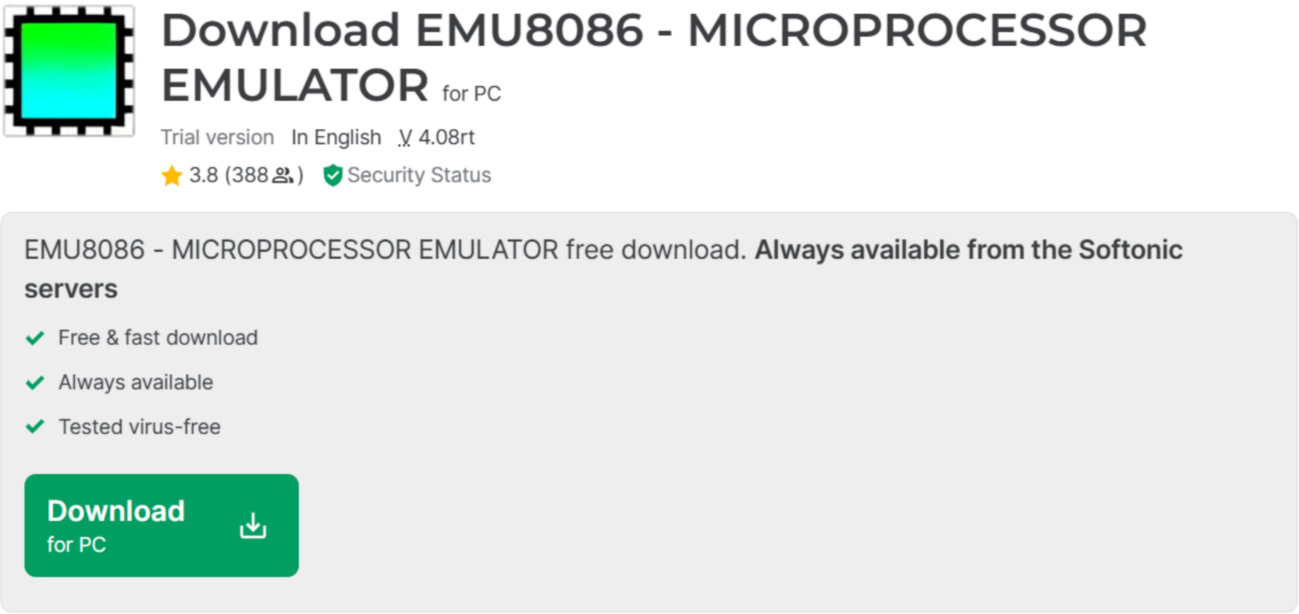
EMU8086 - MICROPROCESSOR EMULATOR

Emulate classic 8086 processors and run vintage software or create custom assembly code effortlessly

Vote: ★★★½☆ (53 votes)
Developer emu8086
Works under: Windows
Program license Trial version
Version 4.08rt

The EMU8086 Microprocessor Emulator is a program that allows users to emulate hardware on Windows PCs.

Click on Download for PC button.



Download EMU8086 - MICROPROCESSOR EMULATOR for PC

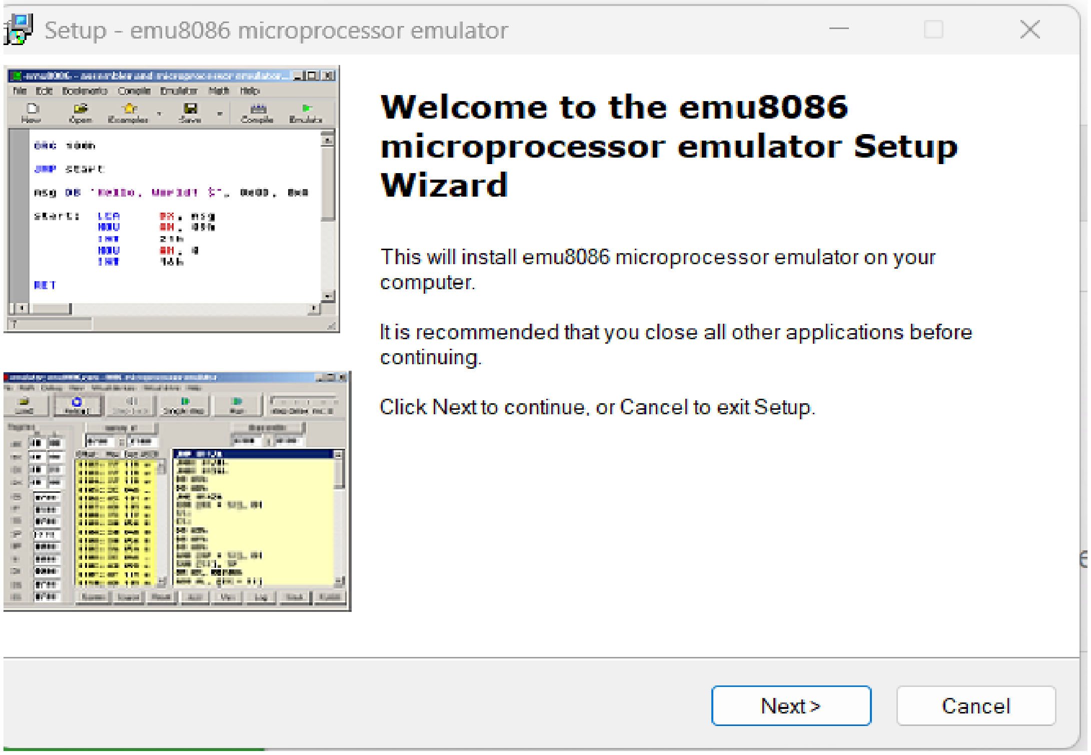
Trial version In English V 4.08rt

★ 3.8 (388) Security Status

EMU8086 - MICROPROCESSOR EMULATOR free download. **Always available from the Softonic servers**

- ✓ Free & fast download
- ✓ Always available
- ✓ Tested virus-free

Step 2: Begin Installation: Double-click the downloaded setup file to launch the installation wizard. Then click on next.



Setup - emu8086 microprocessor emulator

```
ORG 100h

JMP start

msg DB 'Hello, World! $', 0Dh, 0Ah

start:  LEA     DX, msg
        MOV     AH, 09h
        INT     21h
        MOV     AH, 0
        INT     16h

RET
```

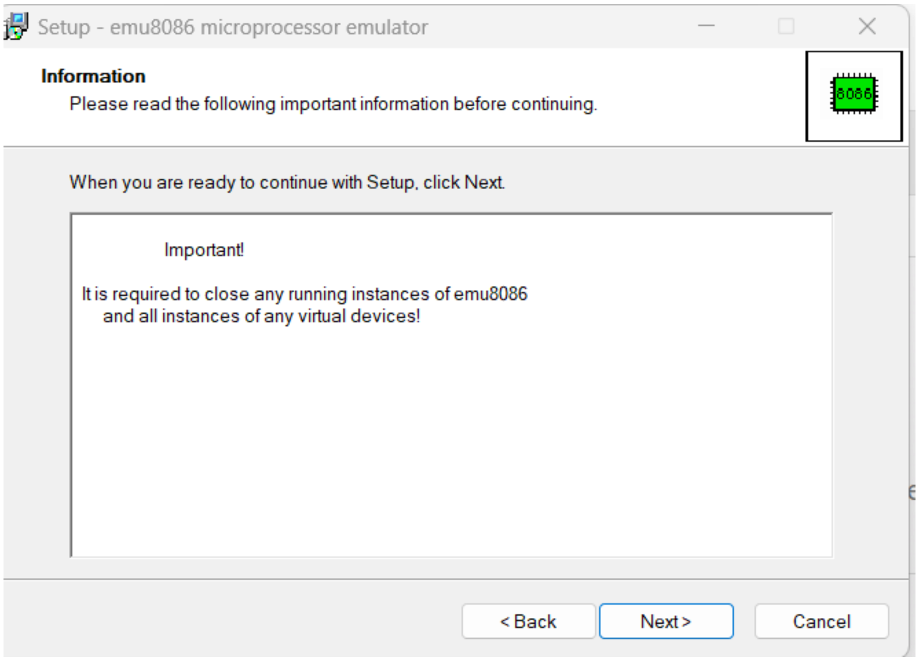
Welcome to the emu8086 microprocessor emulator Setup Wizard

This will install emu8086 microprocessor emulator on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Step 3: Then read the information and click on next.



Information

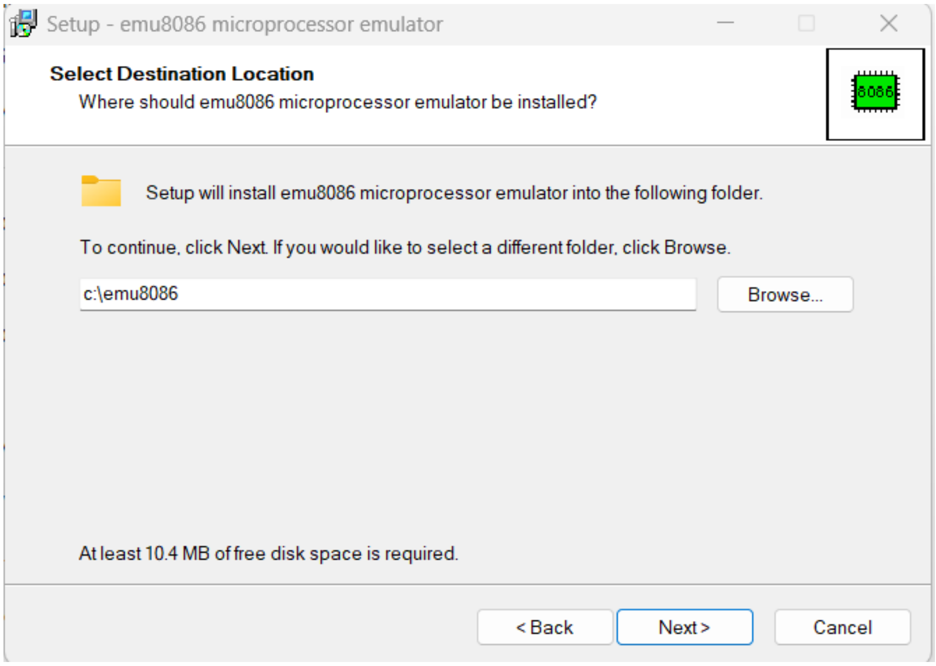
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

Important!

It is required to close any running instances of emu8086
and all instances of any virtual devices!

Step 4: Click on next.



Select Destination Location

Where should emu8086 microprocessor emulator be installed?

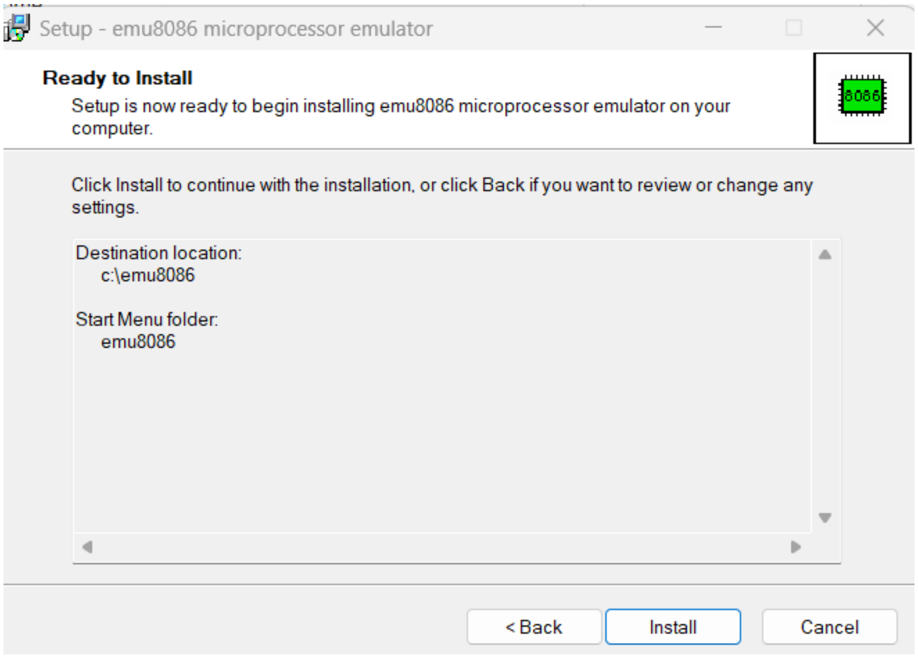
Setup will install emu8086 microprocessor emulator into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

c:\emu8086

At least 10.4 MB of free disk space is required.

Step 5: Then emu8086 microprocessor emulator is ready to installation. Click on install.



Setup - emu8086 microprocessor emulator

Ready to Install

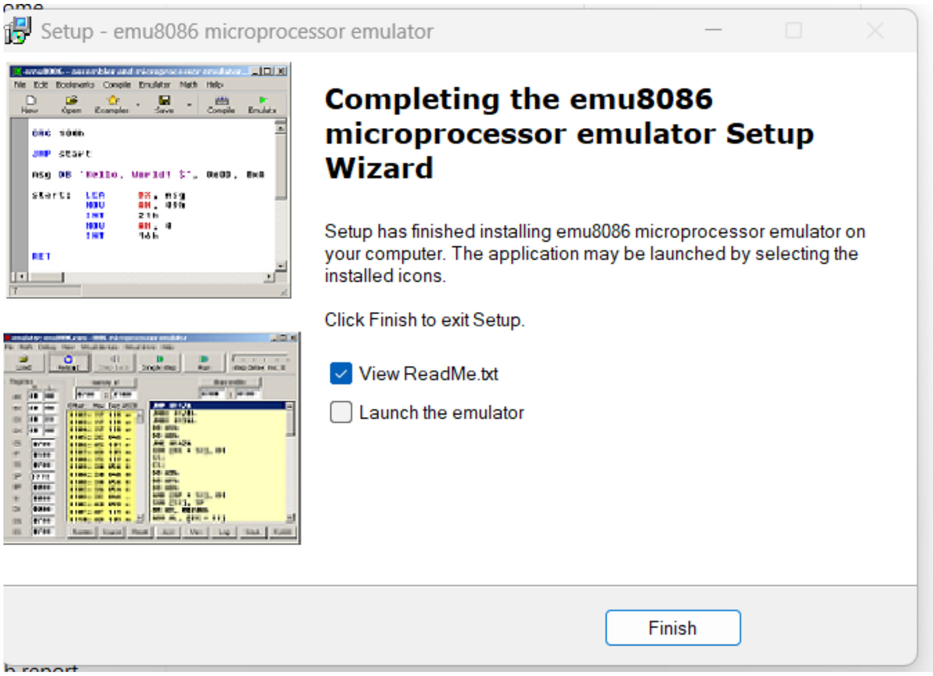
Setup is now ready to begin installing emu8086 microprocessor emulator on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
c:\emu8086

Start Menu folder:
emu8086

Step 6: Click on Finish for completing the emu8086 microprocessor emulator setup wizard.



Setup - emu8086 microprocessor emulator

Completing the emu8086 microprocessor emulator Setup Wizard

Setup has finished installing emu8086 microprocessor emulator on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ View ReadMe.txt

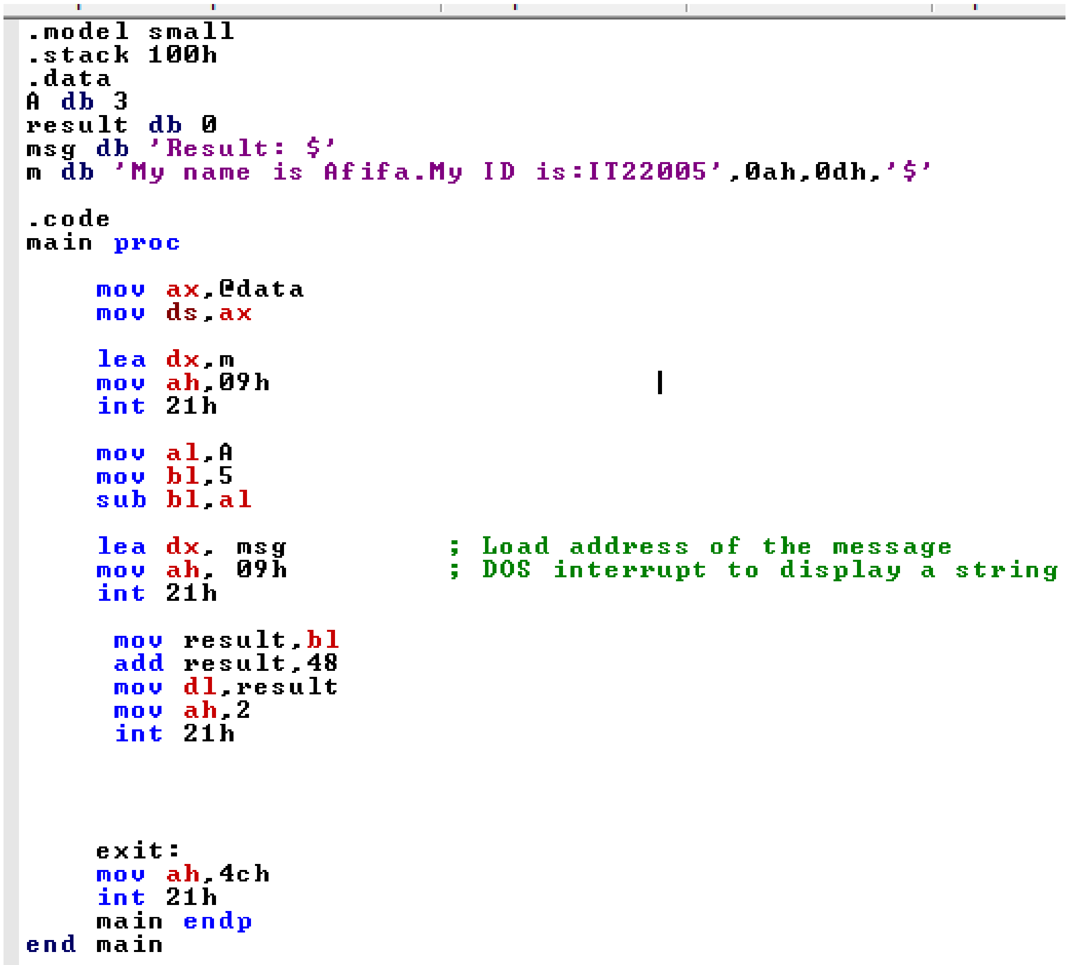
□ Launch the emulator

The installation of emu8086 microprocessor emulator is complete.

Conclusion: The installation of EMU8086 microprocessor emulator was successful, and the software is ready for use in assembly language programming and microprocessor emulation tasks.

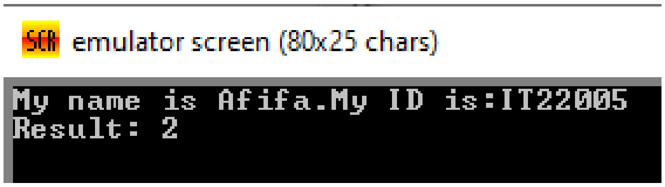
Program 1: A program that takes an input A and computes A = 5 – A

Code:



```
.model small
.stack 100h
.data
A db 3
result db 0
msg db 'Result: $'
m db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'

.code
main proc

    mov ax,@data
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    mov al,A
    mov bl,5
    sub bl,al

    lea dx, msg         ; Load address of the message
    mov ah, 09h         ; DOS interrupt to display a string
    int 21h

     mov result,bl
     add result,48
     mov dl,result
     mov ah,2
     int 21h




    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

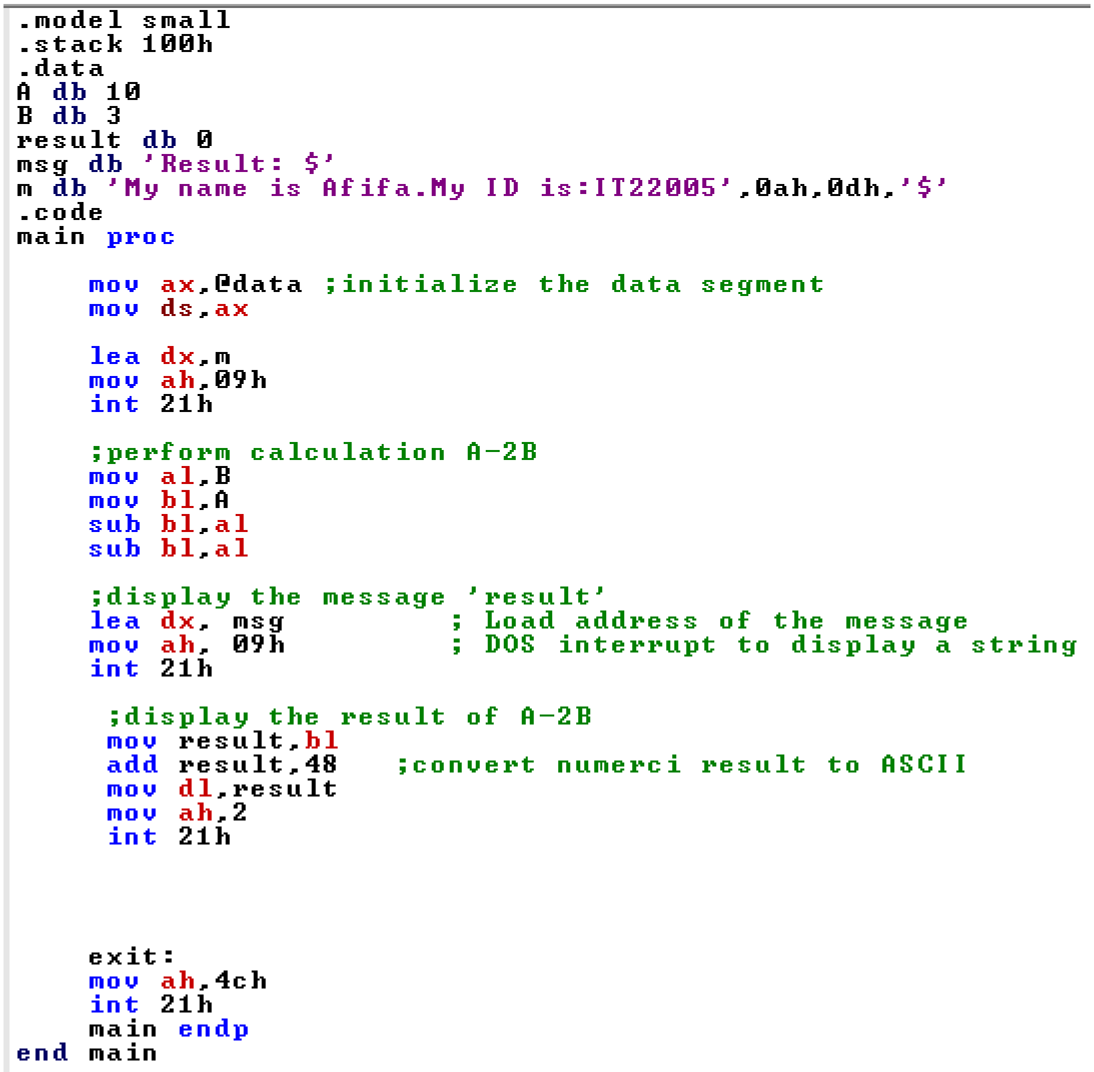
Output:



```
My name is Afifa.My ID is:IT22005
Result: 2
```

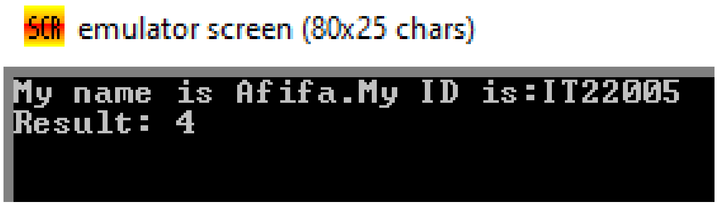
Program 2: A program that takes an input A and computes A = A – 2B

Code:



```
.model small
.stack 100h
.data
A db 10
B db 3
result db 0
msg db 'Result: $'
m db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'
.code
main proc

    mov ax,@data ;initialize the data segment
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    ;perform calculation A-2B
    mov al,B
    mov bl,A
    sub bl,al
    sub bl,al

    ;display the message 'result'
    lea dx, msg         ; Load address of the message
    mov ah, 09h         ; DOS interrupt to display a string
    int 21h

     ;display the result of A-2B
     mov result,bl
     add result,48   ;convert numerci result to ASCII
     mov dl,result
     mov ah,2
     int 21h



    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

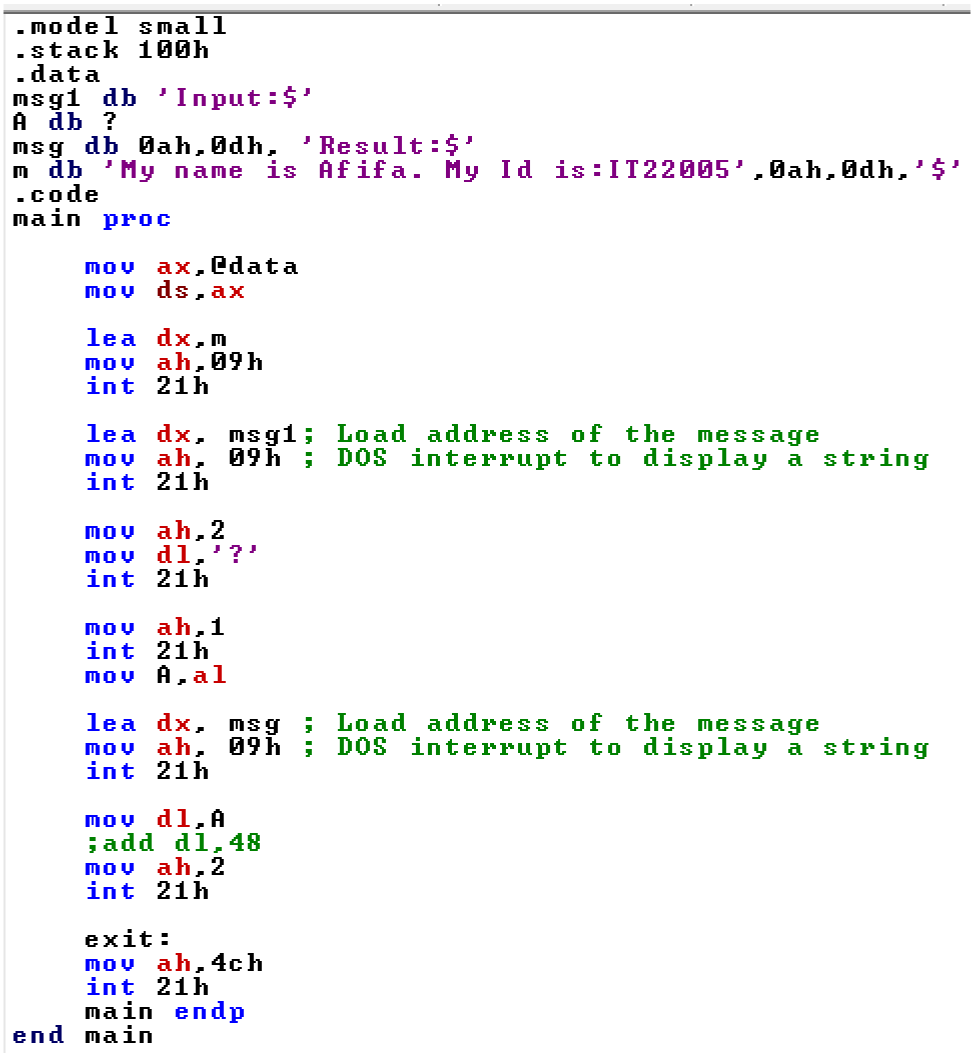
Output:



```
My name is Afifa.My ID is:IT22005
Result: 4
```

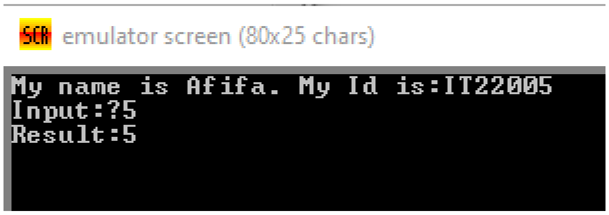
Program 3: A program that shows a question mark (?), takes an input and prints the input text on a new line.

Code:



```
.model small
.stack 100h
.data
msg1 db 'Input:$'
A db ?
msg db 0ah,0dh, 'Result:$'
m db 'My name is Afifa. My Id is:IT22005',0ah,0dh,'$'
.code
main proc

    mov ax,@data
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    lea dx, msg1; Load address of the message
    mov ah, 09h ; DOS interrupt to display a string
    int 21h

    mov ah,2
    mov dl,'?'
    int 21h

    mov ah,1
    int 21h
    mov A,al

    lea dx, msg ; Load address of the message
    mov ah, 09h ; DOS interrupt to display a string
    int 21h

    mov dl,A
    ;add dl,48
    mov ah,2
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

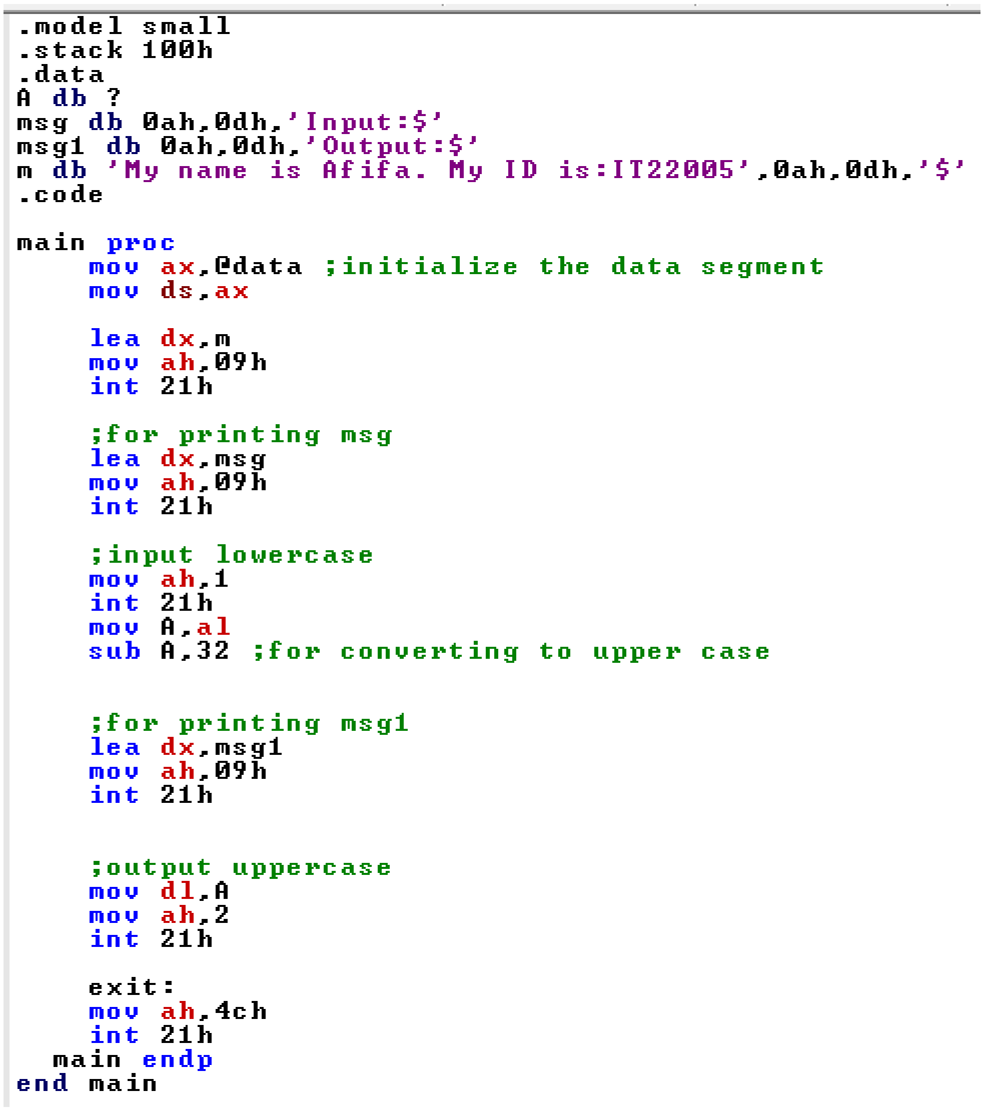
Output:



```
My name is Afifa. My Id is:IT22005
Input:?5
Result:5
```

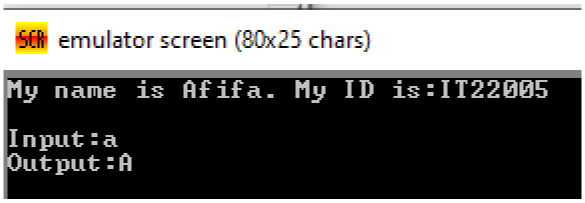
Program 4: A program that takes a lowercase letter and prints it in uppercase.

Code:



```
.model small
.stack 100h
.data
A db ?
msg db 0ah,0dh,'Input:$'
msg1 db 0ah,0dh,'Output:$'
m db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code

main proc
    mov ax,@data ;initialize the data segment
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    ;for printing msg
    lea dx,msg
    mov ah,09h
    int 21h

    ;input lowercase
    mov ah,1
    int 21h
    mov A,al
    sub A,32 ;for converting to upper case


    ;for printing msg1
    lea dx,msg1
    mov ah,09h
    int 21h


    ;output uppercase
    mov dl,A
    mov ah,2
    int 21h

    exit:
    mov ah,4ch
    int 21h
  main endp
end main
```

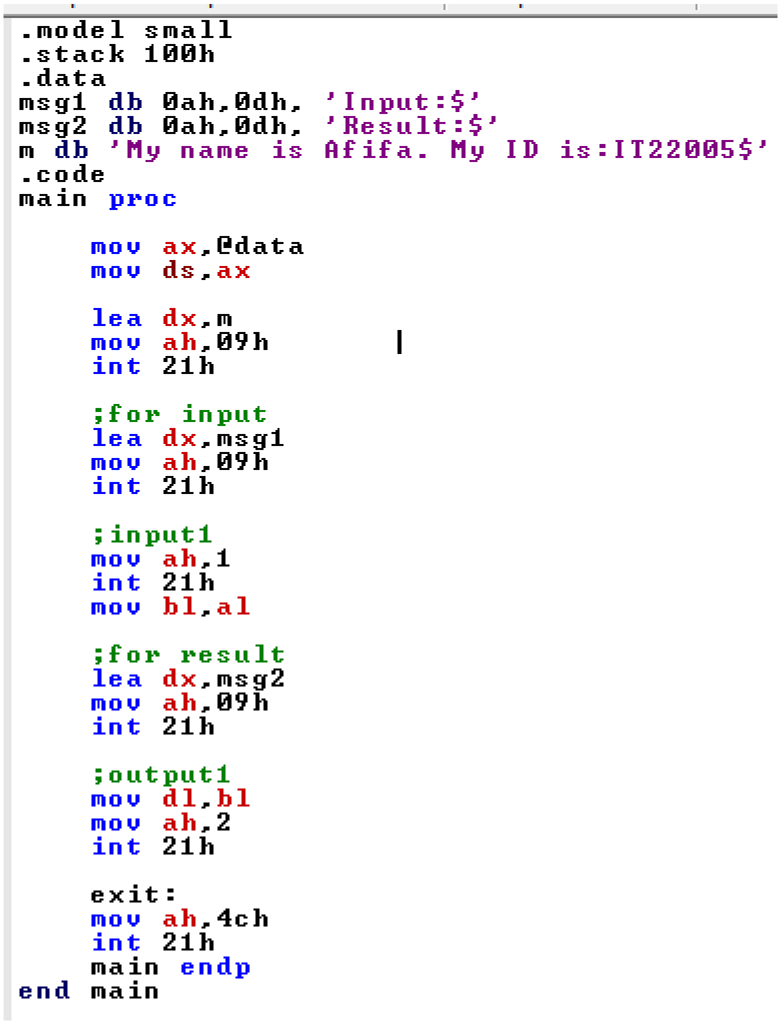
Output:



```
My name is Afifa. My ID is:IT22005

Input:a
Output:A
```

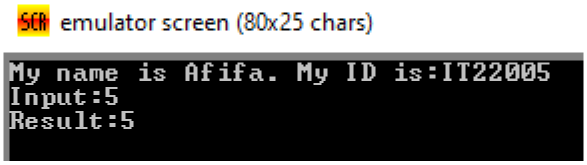
Program 5: Show a message: 'Input:' Then, on the next line, display the message: ‘Result:’

Code:



```
.model small
.stack 100h
.data
msg1 db 0ah,0dh, 'Input:$'
msg2 db 0ah,0dh, 'Result:$'
m db 'My name is Afifa. My ID is:IT22005$'
.code
main proc

     mov ax,@data
     mov ds,ax

     lea dx,m
     mov ah,09h
     int 21h

     ;for input
     lea dx,msg1
     mov ah,09h
     int 21h

     ;input1
     mov ah,1
     int 21h
     mov bl,al

     ;for result
     lea dx,msg2
     mov ah,09h
     int 21h

     ;output1
     mov dl,bl
     mov ah,2
     int 21h

     exit:
     mov ah,4ch
     int 21h
     main endp
end main
```

Output:



```
My name is Afifa. My ID is:IT22005
Input:5
Result:5
```

Conclusion: This lab provided hands-on experience with basic assembly operations. The tasks demonstrated the use of arithmetic, input/output, and character manipulation, enhancing our understanding of low-level programming. Future improvements may include more complex operations or the integration of loops and conditional branches for extended functionality.

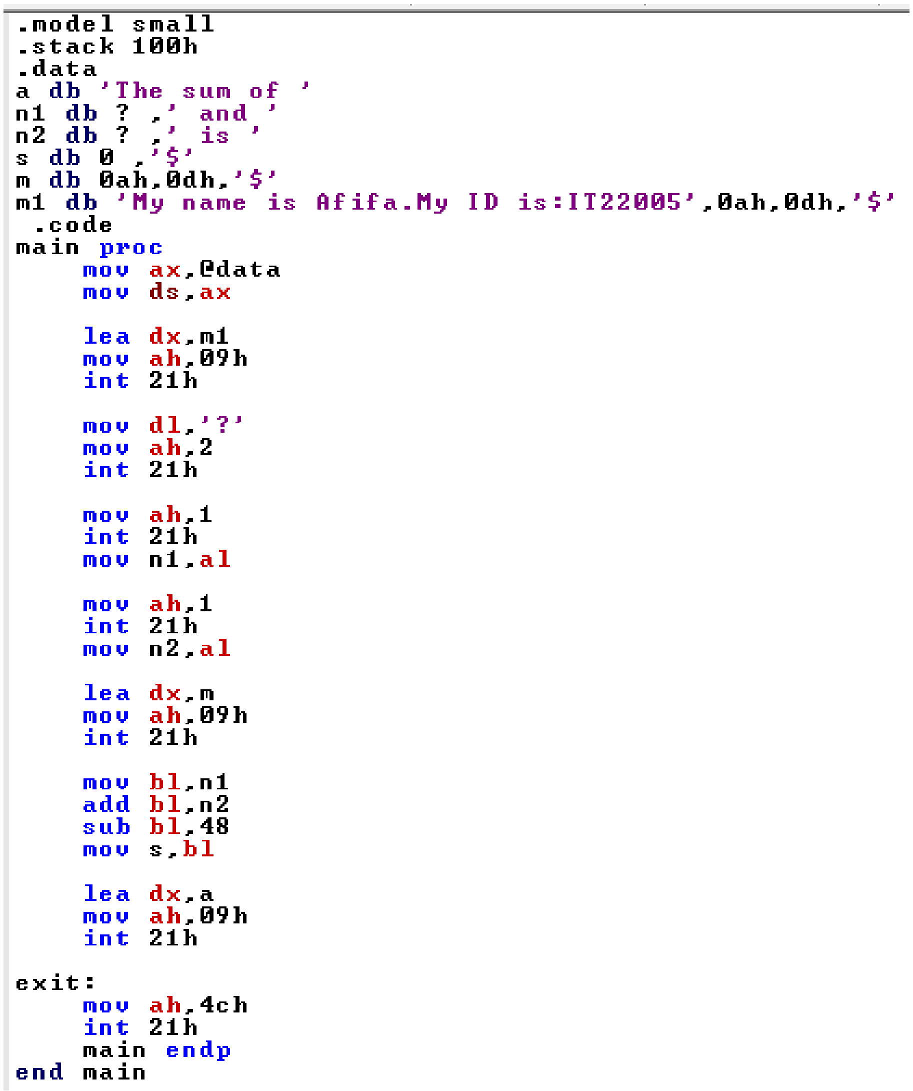
Program 1: A program to (a) display a "?", (b) read two decimal digits whose sum is less than 10, (c) display them and their sum on the next line, with an appropriate message.

Sample execution:

?27

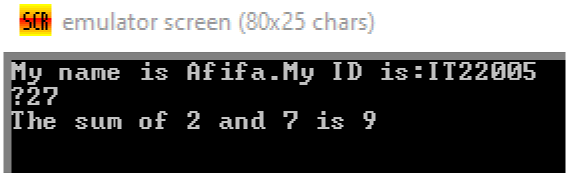
THE SUM OF 2 AND 7 IS 9

Code:



```
.model small
.stack 100h
.data
a db 'The sum of '
n1 db ? ,' and '
n2 db ? ,' is '
s db 0 ,'$'
m db 0ah,0dh,'$'
m1 db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'
 .code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,m1
    mov ah,09h
    int 21h

    mov dl,'?'
    mov ah,2
    int 21h

    mov ah,1
    int 21h
    mov n1,al

    mov ah,1
    int 21h
    mov n2,al

    lea dx,m
    mov ah,09h
    int 21h

    mov bl,n1
    add bl,n2
    sub bl,48
    mov s,bl

    lea dx,a
    mov ah,09h
    int 21h

exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

Output:



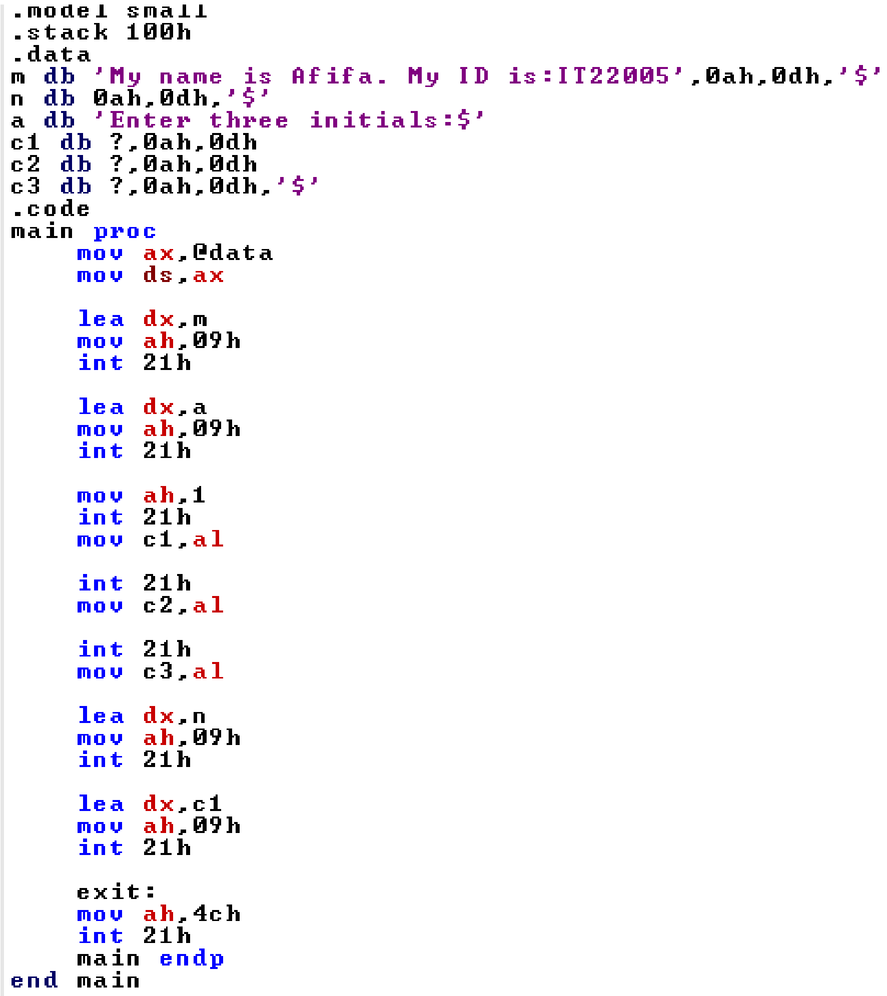
```
My name is Afifa.My ID is:IT22005
?27
The sum of 2 and 7 is 9
```

Program 2: A program to (a) prompt the user, (b) read first, middle, and last initials of a person's name, and (c) display them down the left margin.

Sample execution:

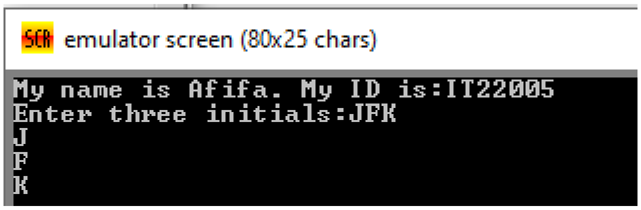
ENTER THREE INITIALS: JFK
J
F
K

Code:



```
.model small
.stack 100h
.data
m db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
n db 0ah,0dh,'$'
a db 'Enter three initials:$'
c1 db ?,0ah,0dh
c2 db ?,0ah,0dh
c3 db ?,0ah,0dh,'$'
.code
main proc
        mov ax,@data
        mov ds,ax

        lea dx,m
        mov ah,09h
        int 21h

        lea dx,a
        mov ah,09h
        int 21h

        mov ah,1
        int 21h
        mov c1,al

        int 21h
        mov c2,al

        int 21h
        mov c3,al

        lea dx,n
        mov ah,09h
        int 21h

        lea dx,c1
        mov ah,09h
        int 21h

        exit:
        mov ah,4ch
        int 21h
        main endp
end main
```

Output:



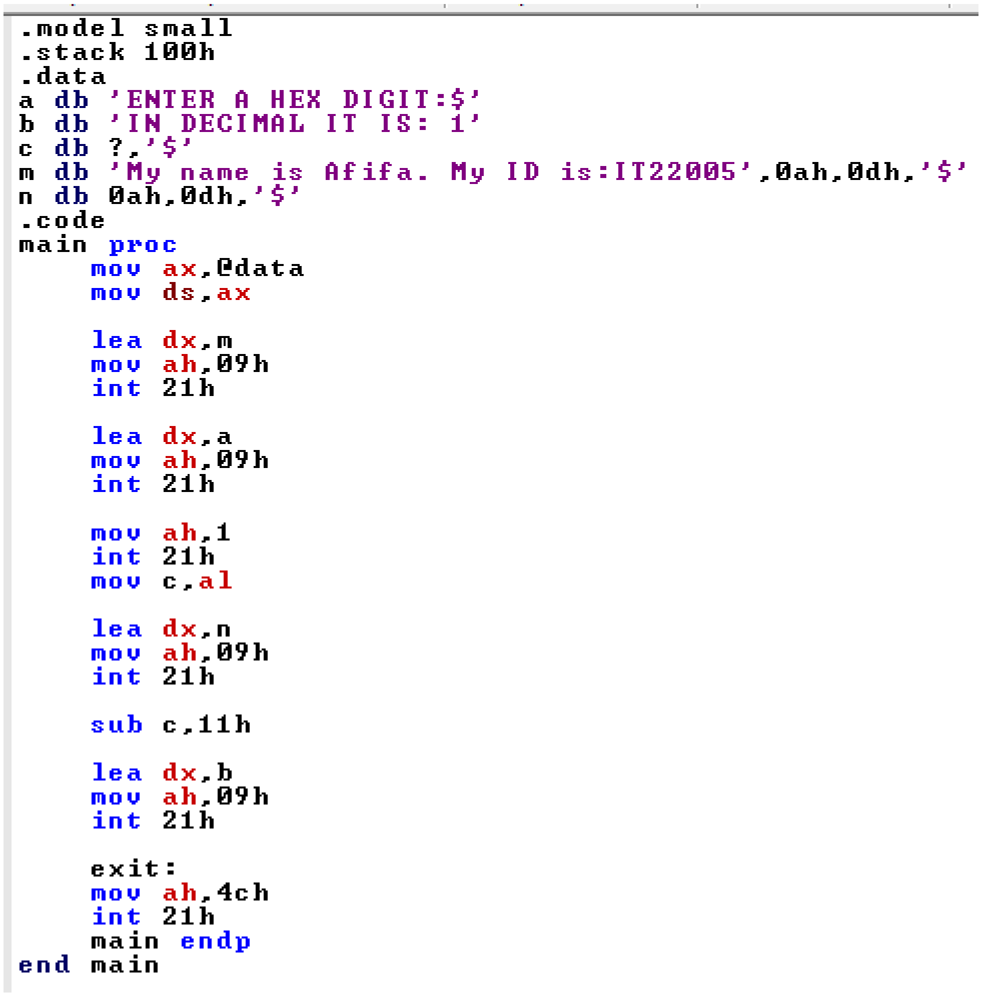
```
My name is Afifa. My ID is:IT22005
Enter three initials:JFK
J
F
K
```

Program 3: A program to read one of the hex digits A-F, and display it on the next line in decimal.

ENTER A HEX DIGIT: C

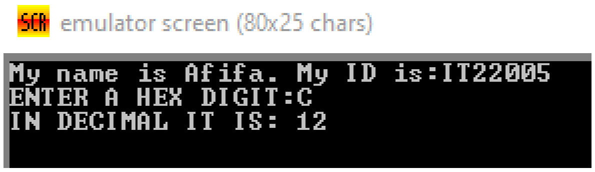
IN DECIMAL IT IS: 12

Code:



```
.model small
.stack 100h
.data
a db 'ENTER A HEX DIGIT:$'
b db 'IN DECIMAL IT IS: 1'
c db ?,'$'
m db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
n db 0ah,0dh,'$'
.code
main proc
     mov ax,@data
     mov ds,ax

     lea dx,m
     mov ah,09h
     int 21h

     lea dx,a
     mov ah,09h
     int 21h

     mov ah,1
     int 21h
     mov c,al

     lea dx,n
     mov ah,09h
     int 21h

     sub c,11h

     lea dx,b
     mov ah,09h
     int 21h

     exit:
     mov ah,4ch
     int 21h
     main endp
end main
```

Output:

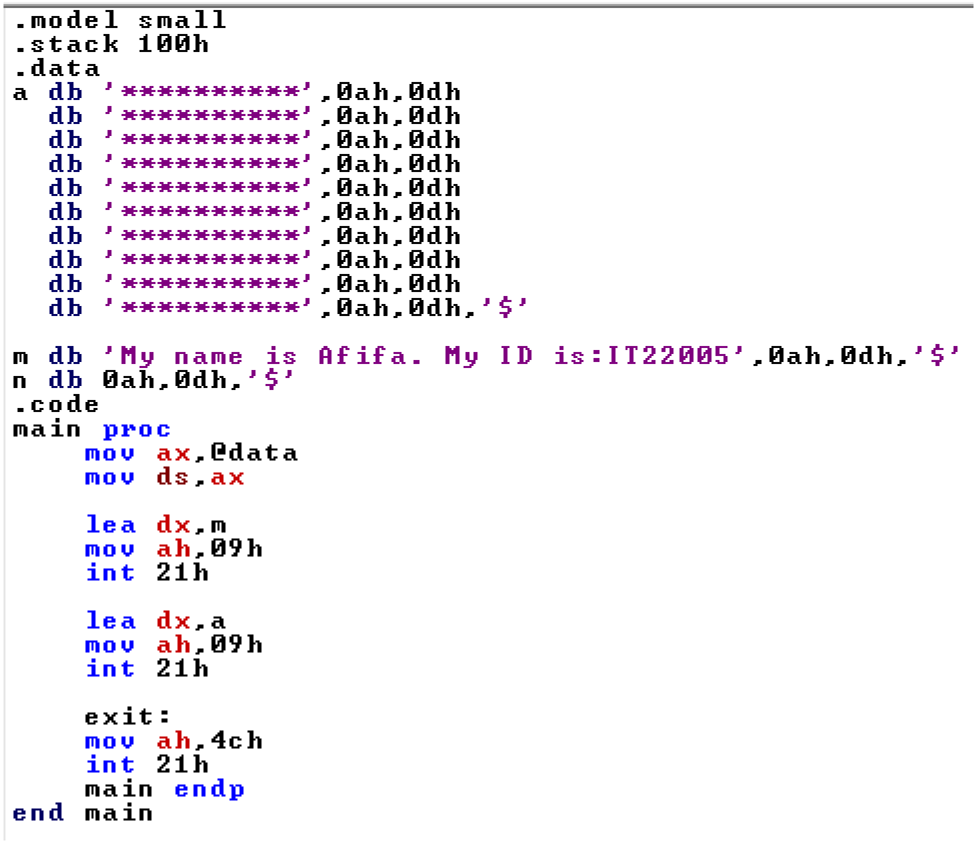


```
My name is Afifa. My ID is:IT22005
ENTER A HEX DIGIT:C
IN DECIMAL IT IS: 12
```

Program 4: A program to display a 10 x 10 solid box of asterisks.

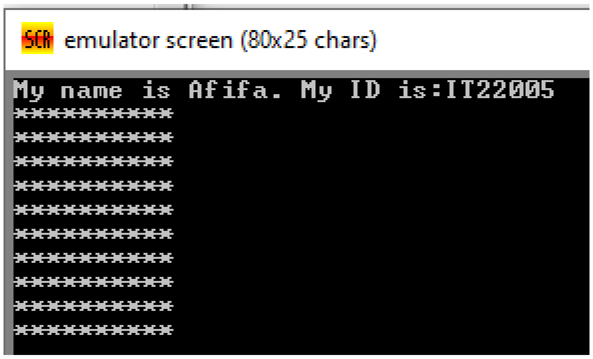
Hint: declare a string in the data segment that specifies the box, and display it with INT 2lh, function 9h.

Code:



```
.model small
.stack 100h
.data
a db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh
  db '**********',0ah,0dh,'$'

m db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
n db 0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    lea dx,a
    mov ah,09h
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

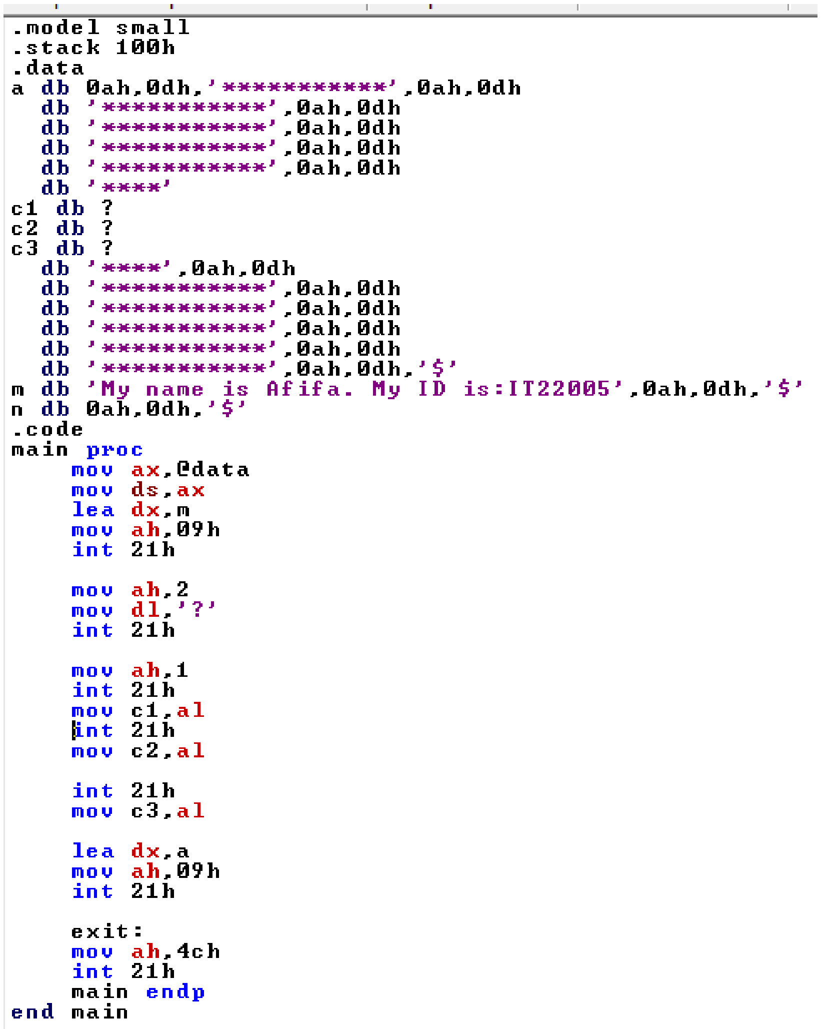
Output:



```
My name is Afifa. My ID is:IT22005
**********
**********
**********
**********
**********
**********
**********
**********
**********
**********
```

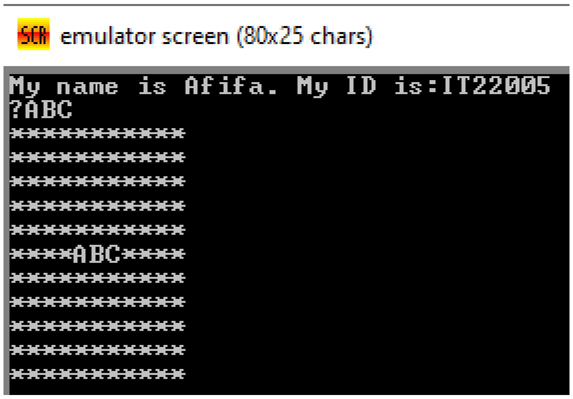
Program 5: A program to (a) display"?", (b) read three initials,(c) display them in the middle of an 11 x 11 box of asteriks, and (d) beep the computer.

Code:



```
.model small
.stack 100h
.data
a db 0ah,0dh,'***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '****'
c1 db ?
c2 db ?
c3 db ?
  db '****',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh
  db '***********',0ah,0dh,'$'
m db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
n db 0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax
    lea dx,m
    mov ah,09h
    int 21h

    mov ah,2
    mov dl,'?'
    int 21h

    mov ah,1
    int 21h
    mov c1,al
    int 21h
    mov c2,al

    int 21h
    mov c3,al

    lea dx,a
    mov ah,09h
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

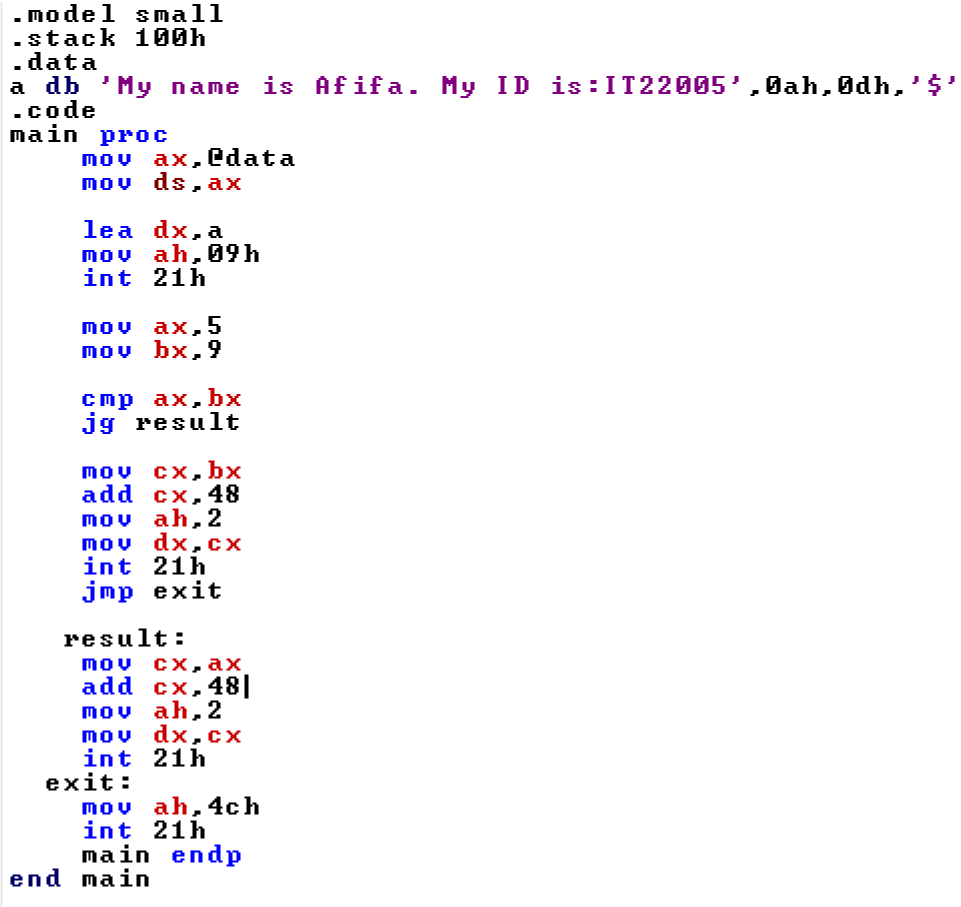
Output:



```
My name is Afifa. My ID is:IT22005
?ABC
***********
***********
***********
***********
***********
****ABC****
***********
***********
***********
***********
***********
```

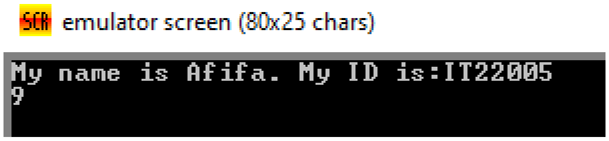
Program 1: Ax and Bx contain signed numbers. A program to put the biggest one in Cx.

Code:



```
.model small
.stack 100h
.data
a db ’My name is Afifa. My ID is:IT22005’,0ah,0dh,’$’
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ax,5
    mov bx,9

    cmp ax,bx
    jg result

    mov cx,bx
    add cx,48
    mov ah,2
    mov dx,cx
    int 21h
    jmp exit

   result:
    mov cx,ax
    add cx,48
    mov ah,2
    mov dx,cx
    int 21h
  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

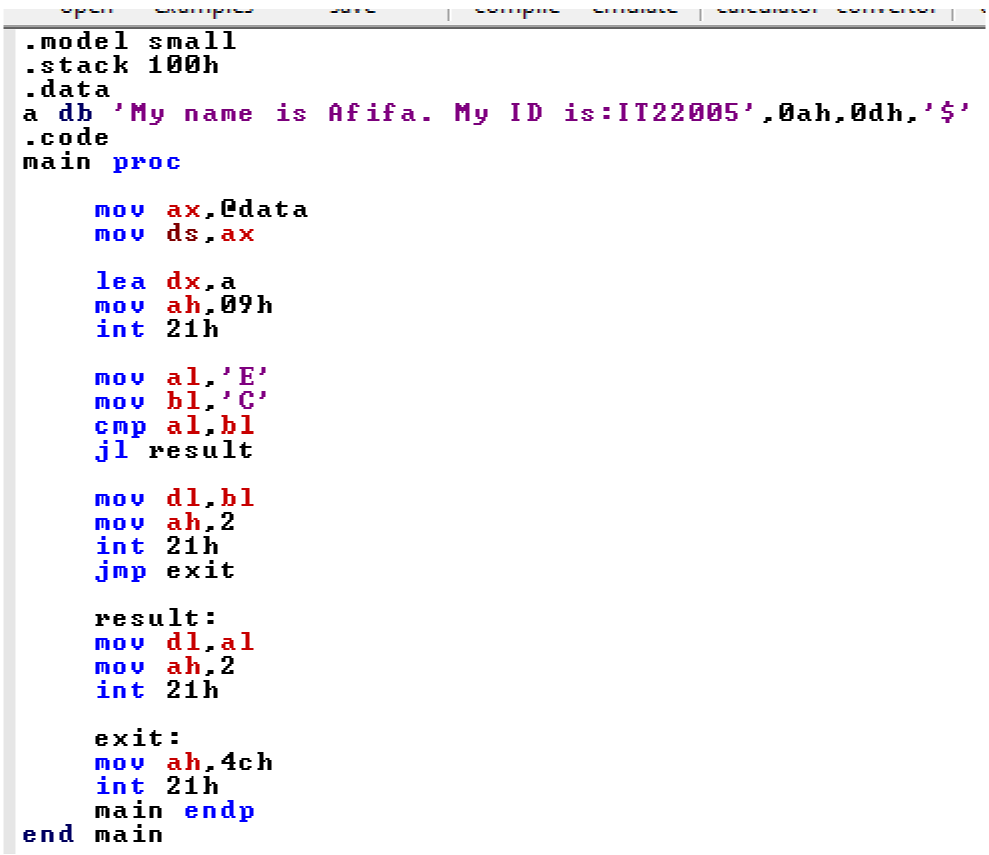
Output:



```
My name is Afifa. My ID is:IT22005
9
```

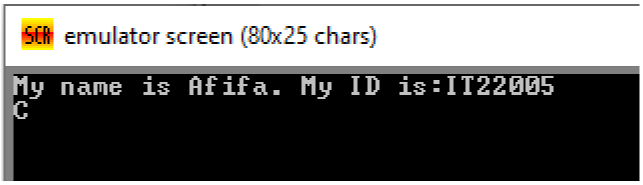
Program 2: AL and BL contain extended ASCII characters. A program to display the one that comes first in the character sequence.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc

    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov al,'E'
    mov bl,'C'
    cmp al,bl
    jl result

    mov dl,bl
    mov ah,2
    int 21h
    jmp exit

    result:
    mov dl,al
    mov ah,2
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

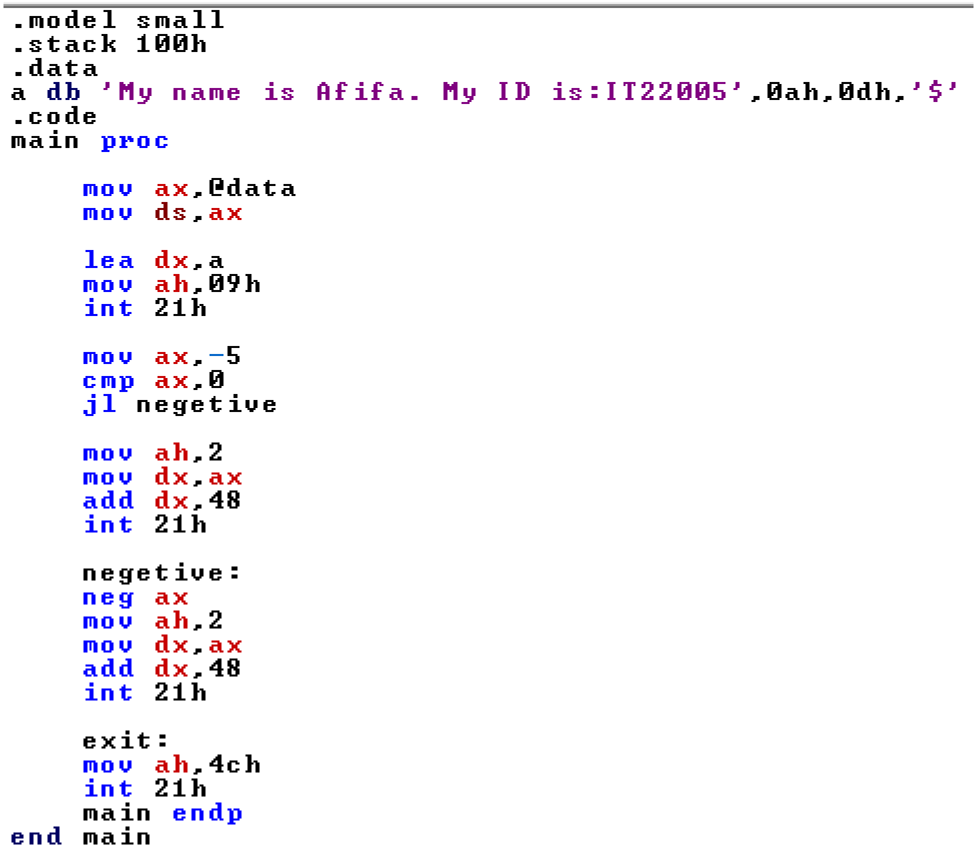
Output:



```
My name is Afifa. My ID is:IT22005
C
```

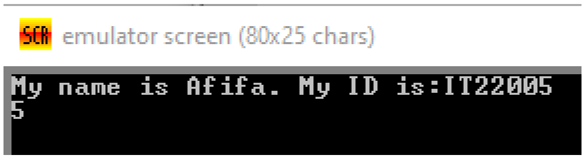
Program 3: Replace the number in AX by its absolute value.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc

    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ax,-5
    cmp ax,0
    jl negetive

    mov ah,2
    mov dx,ax
    add dx,48
    int 21h

    negetive:
    neg ax
    mov ah,2
    mov dx,ax
    add dx,48
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

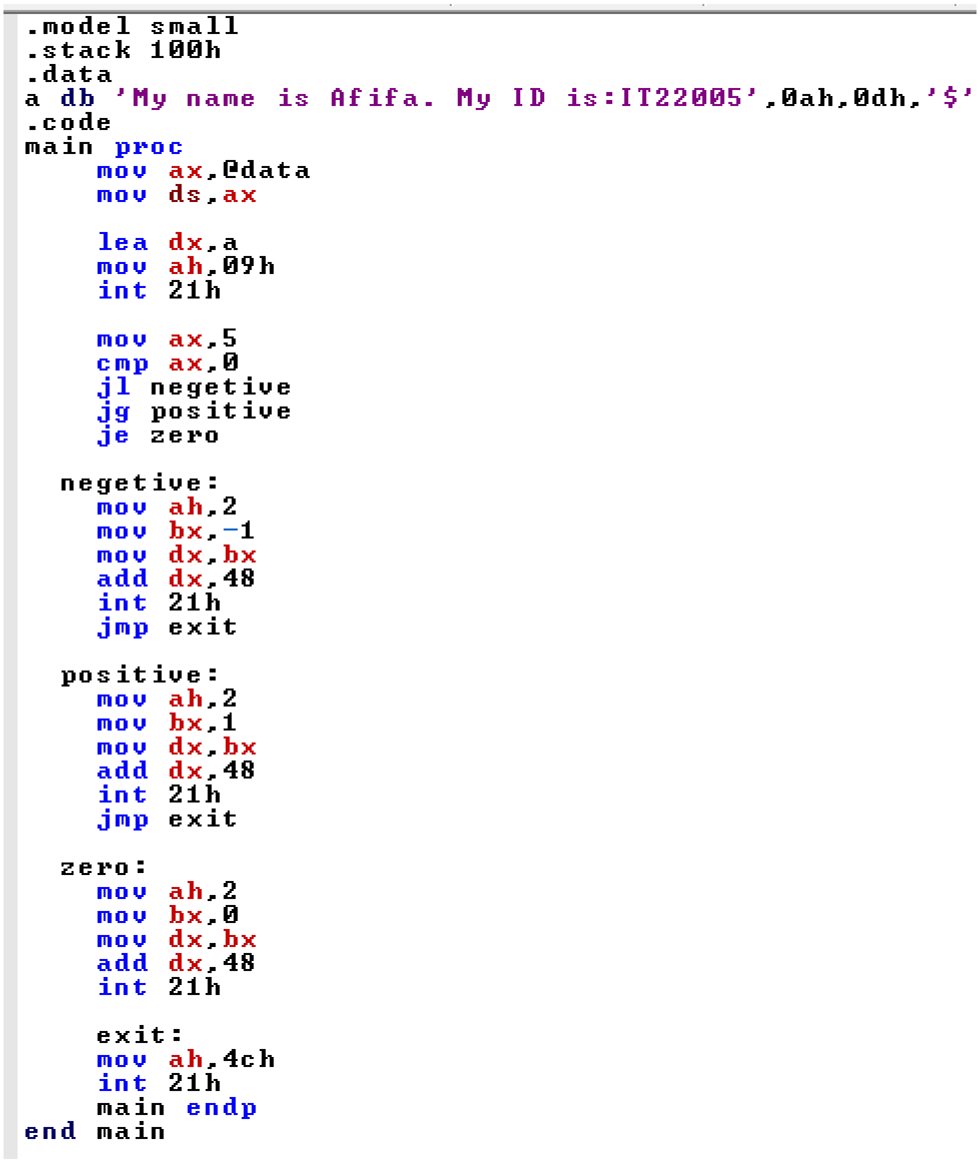
Output:



```
My name is Afifa. My ID is:IT22005
5
```

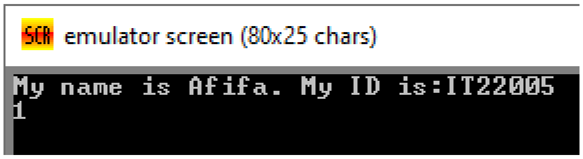
Program 4: If AX contains a negative number, put -1 In BX; if AX contains 0, put O In BX; if AX contains a positive number, put 1 In BX.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ax,5
    cmp ax,0
    jl negetive
    jg positive
    je zero

  negetive:
    mov ah,2
    mov bx,-1
    mov dx,bx
    add dx,48
    int 21h
    jmp exit

  positive:
    mov ah,2
    mov bx,1
    mov dx,bx
    add dx,48
    int 21h
    jmp exit

  zero:
    mov ah,2
    mov bx,0
    mov dx,bx
    add dx,48
    int 21h

    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

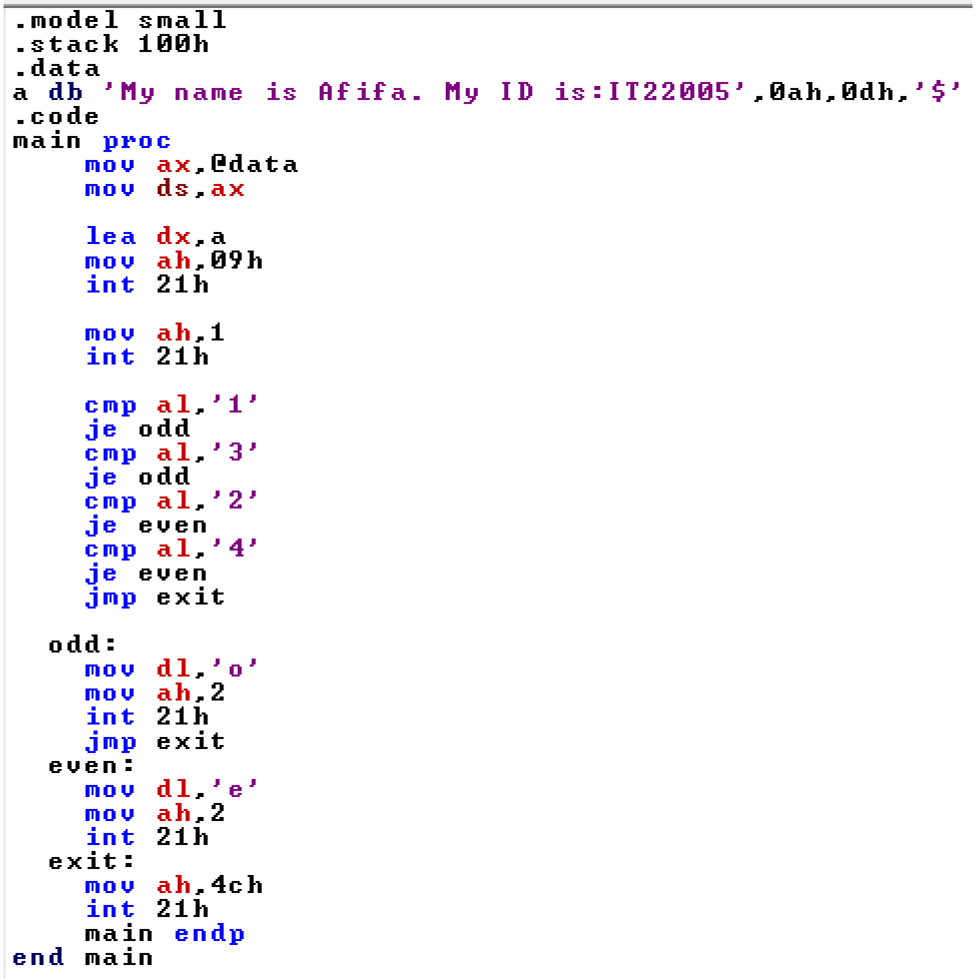
Output:



```
My name is Afifa. My ID is:IT22005
1
```

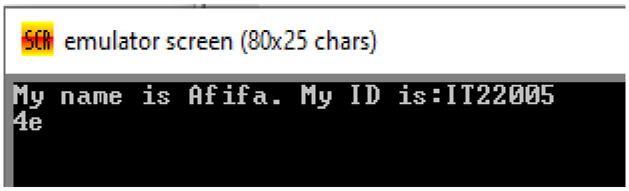
Program 5: A program which display “o” if AL contains 1 or 3; if AL contains 2 or 4, display "e".

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ah,1
    int 21h

    cmp al,'1'
    je odd
    cmp al,'3'
    je odd
    cmp al,'2'
    je even
    cmp al,'4'
    je even
    jmp exit

  odd:
    mov dl,'o'
    mov ah,2
    int 21h
    jmp exit
  even:
    mov dl,'e'
    mov ah,2
    int 21h
  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

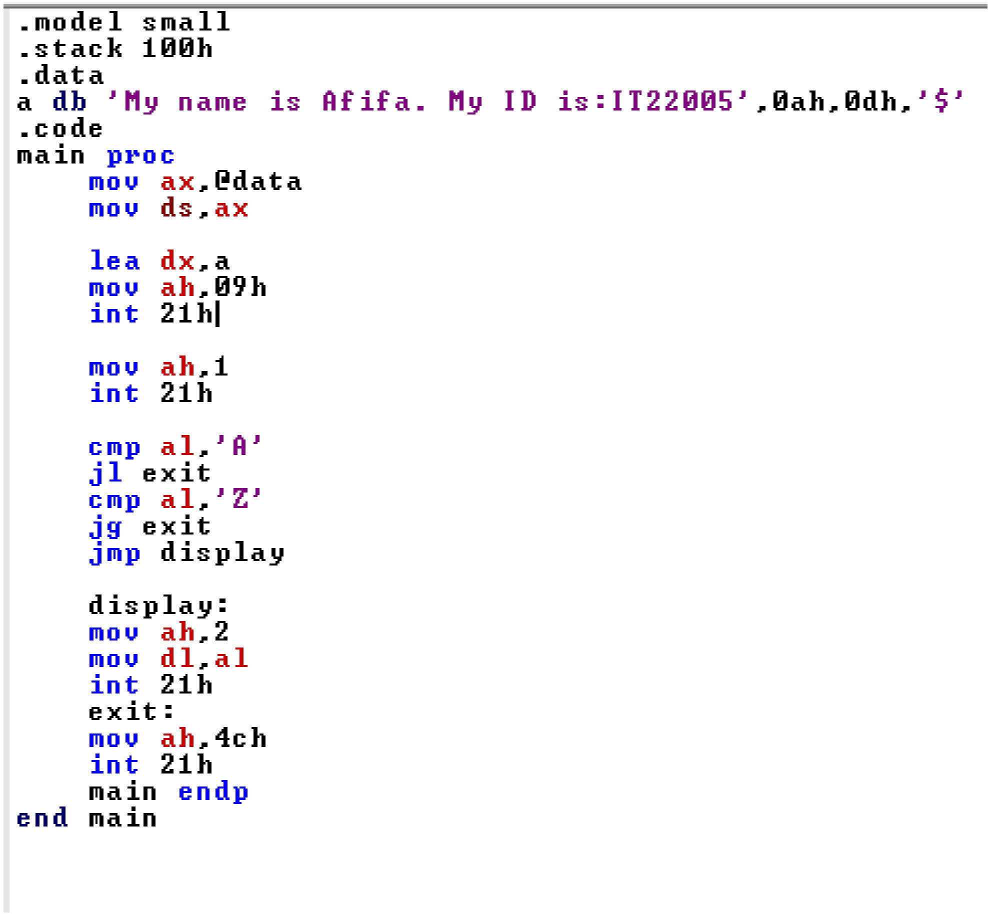
Output:



```
My name is Afifa. My ID is:IT22005
4e
```

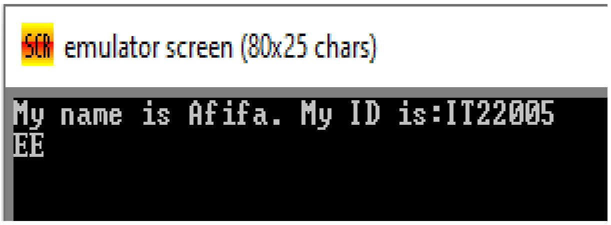
Program 6: A program which read a character, and if it's an uppercase letter, display it.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ah,1
    int 21h

    cmp al,'A'
    jl exit
    cmp al,'Z'
    jg exit
    jmp display

    display:
    mov ah,2
    mov dl,al
    int 21h
    exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

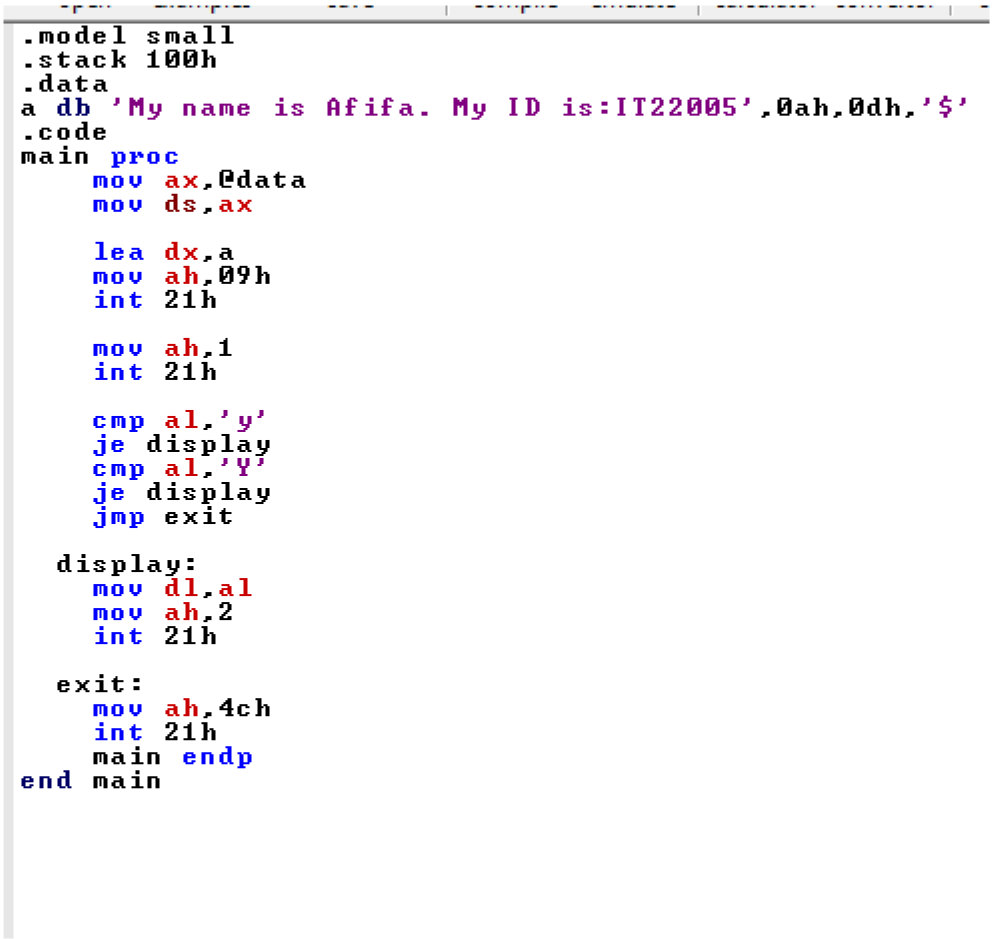
Output:



```
My name is Afifa. My ID is:IT22005
EE
```

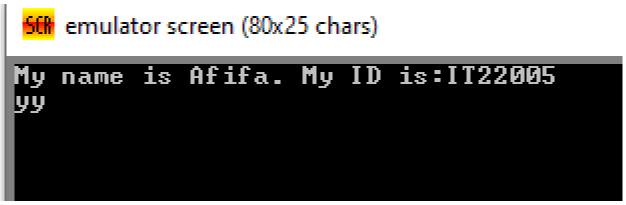
Program 7: A program which read a character. If it's "y" or "Y", display it; otherwise terminate the program.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov ah,1
    int 21h

    cmp al,'y'
    je display
    cmp al,'Y'
    je display
    jmp exit

  display:
    mov dl,al
    mov ah,2
    int 21h

  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

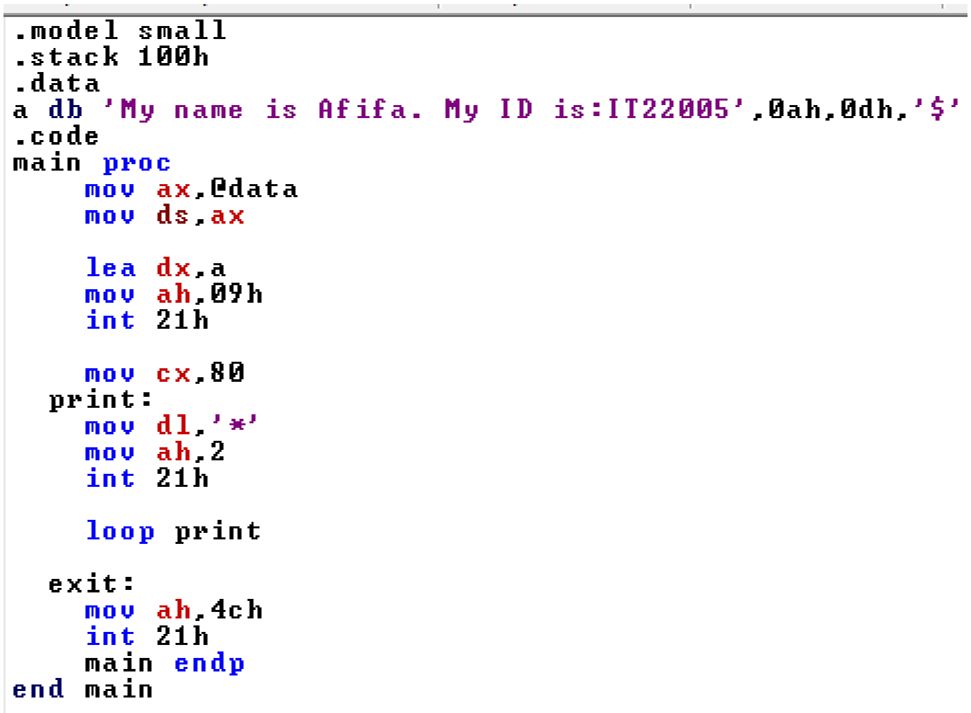
Output:



```
My name is Afifa. My ID is:IT22005
yy
```

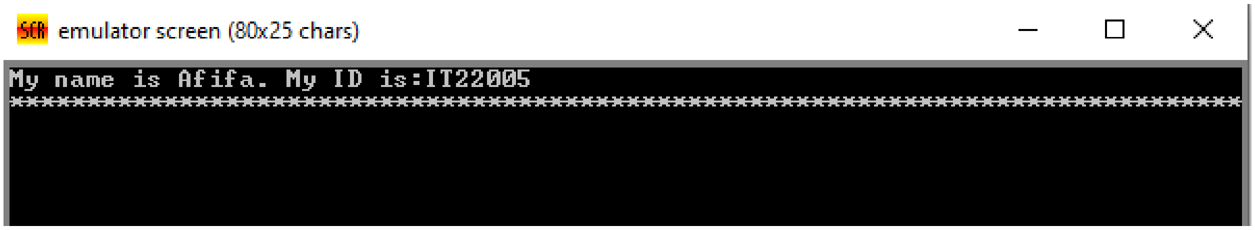
Program 8: A program to display a row of 80 stars by using count-controlled loop.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov cx,80
  print:
    mov dl,'*'
    mov ah,2
    int 21h

    loop print

  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

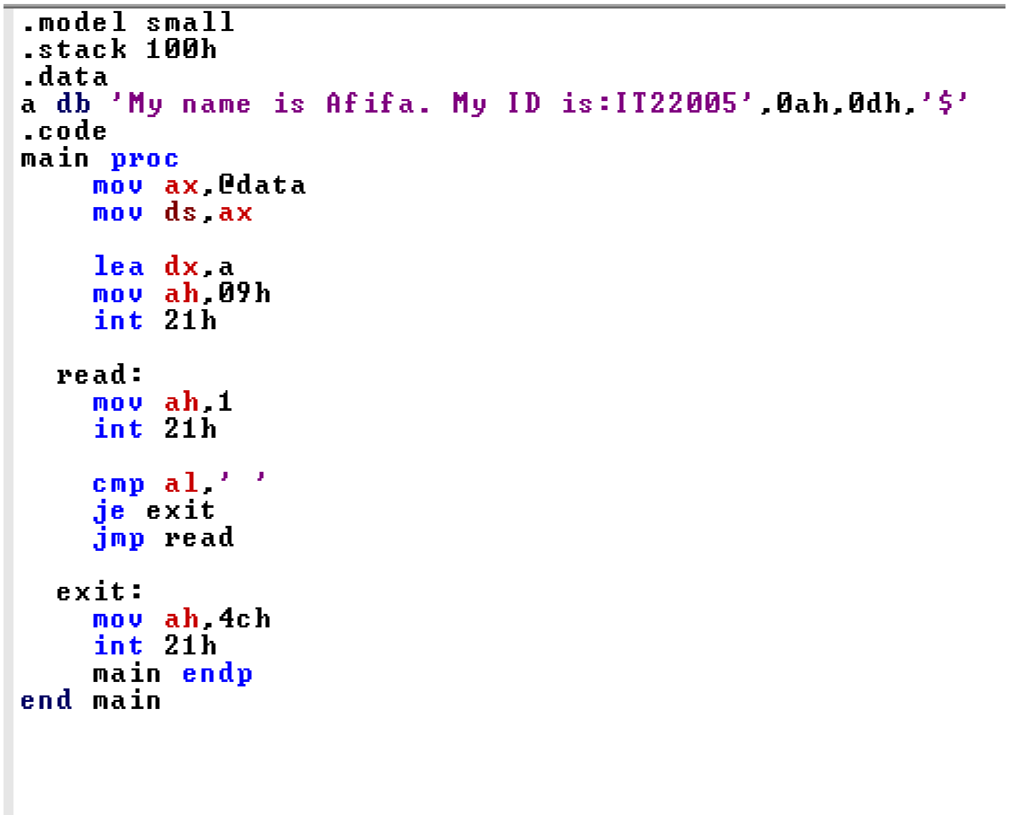
Output:



```
My name is Afifa. My ID is:IT22005
********************************************************************************
```

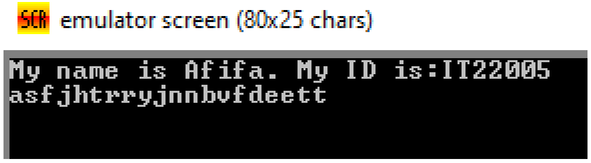
Program 9: Write some code to read characters until a blank is read.

Code:



```
.model small
.stack 100h
.data
a db 'My name is Afifa. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

  read:
    mov ah,1
    int 21h

    cmp al,' '
    je exit
    jmp read

  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

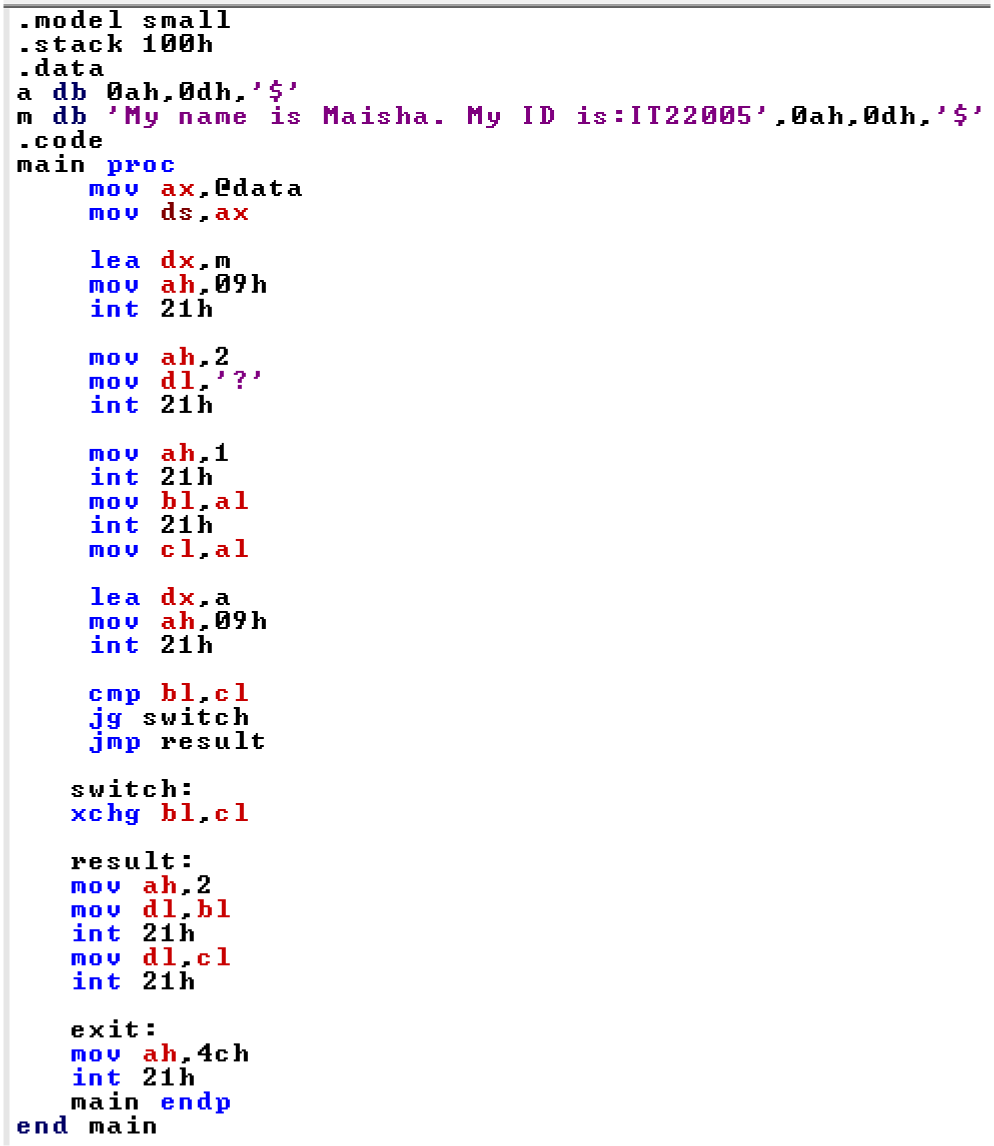
Output:



```
My name is Afifa. My ID is:IT22005
asfjhtrryjnnbvfdeett
```

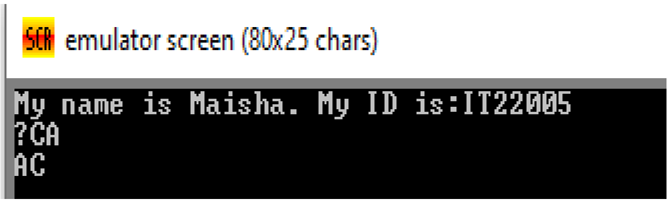
Program 1: Write a program to display a "?", read two capital letters, and display them on the next line in alphabetical order.

Code:



```
.model small
.stack 100h
.data
a db 0ah,0dh,'$'
m db 'My name is Maisha. My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,m
    mov ah,09h
    int 21h

    mov ah,2
    mov dl,'?'
    int 21h

    mov ah,1
    int 21h
    mov bl,al
    int 21h
    mov cl,al

    lea dx,a
    mov ah,09h
    int 21h

    cmp bl,cl
    jg switch
    jmp result

   switch:
   xchg bl,cl

   result:
   mov ah,2
   mov dl,bl
   int 21h
   mov dl,cl
   int 21h

   exit:
   mov ah,4ch
   int 21h
   main endp
end main
```

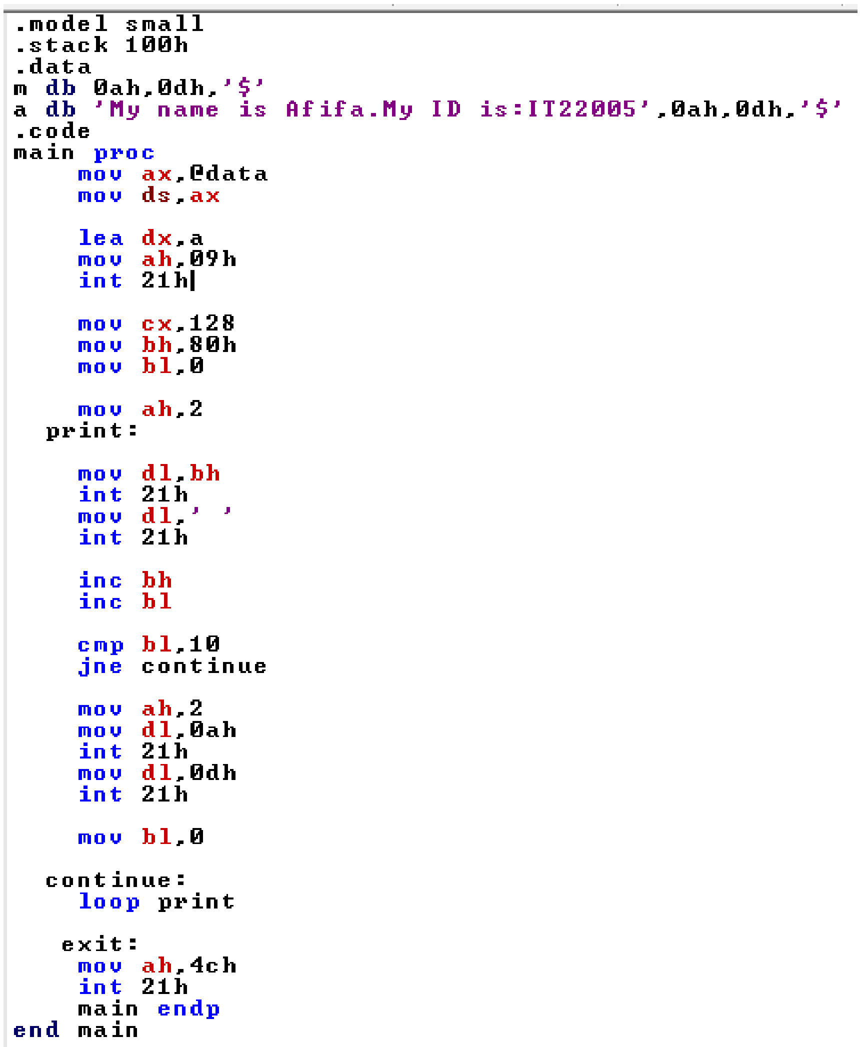
Output:



```
My name is Maisha. My ID is:IT22005
?CA
AC
```

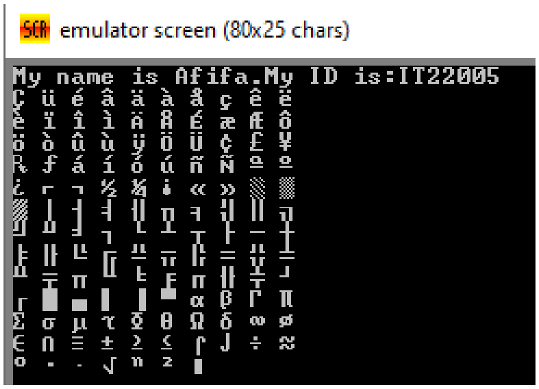
Program 2: A program to display the extended ASCII characters (ASCII codes 80h to FFh). Display 10 characters per line, separated by blanks. Stop after the extended characters have been displayed once.

Code:



```
.model small
.stack 100h
.data
m db 0ah,0dh,'$'
a db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

    mov cx,128
    mov bh,80h
    mov bl,0

    mov ah,2
  print:

    mov dl,bh
    int 21h
    mov dl,' '
    int 21h

    inc bh
    inc bl

    cmp bl,10
    jne continue

    mov ah,2
    mov dl,0ah
    int 21h
    mov dl,0dh
    int 21h

    mov bl,0

  continue:
    loop print

   exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

Output:



```
My name is Afifa.My ID is:IT22005
Ç ü é â ä à å ç ê ë
è ï î ì Ä Å É æ Æ ô
ö ò û ù ÿ Ö Ü ¢ £ ¥
₧ ƒ á í ó ú ñ Ñ ª º
¿ ⌐ ¬ ½ ¼ ¡ « » ░ ▒
▓ │ ┤ ╡ ╢ ╖ ╕ ╣ ║ ╗
╝ ╜ ╛ ┐ └ ┴ ┬ ├ ─ ┼
╞ ╟ ╚ ╔ ╩ ╦ ╠ ═ ╬ ╧
╨ ╤ ╥ ╙ ╘ ╒ ╓ ╫ ╪ ┘
┌ █ ▄ ▌ ▐ ▀ α ß Γ π
Σ σ µ τ Φ Θ Ω δ ∞ φ
ε ∩ ≡ ± ≥ ≤ ⌠ ⌡ ÷ ≈
° ∙ · √ ⁿ ² ■
```

Program 3: A program that will prompt the user to enter a hex digit character ("0"· ... "9" or "A" ... "F"), display it on the next line in decimal, and ask the user if he or she wants to do it again. If the user types "y" or "Y", the program repeats; If the user types anything else, the program terminates. If the user enters an illegal character, prompt the user to try again.

Sample execution:

ENTER A HEX DIGIT: 9

IN DECIMAL IS IT 9

DO YOU WANT TO DO IT AGAIN? y

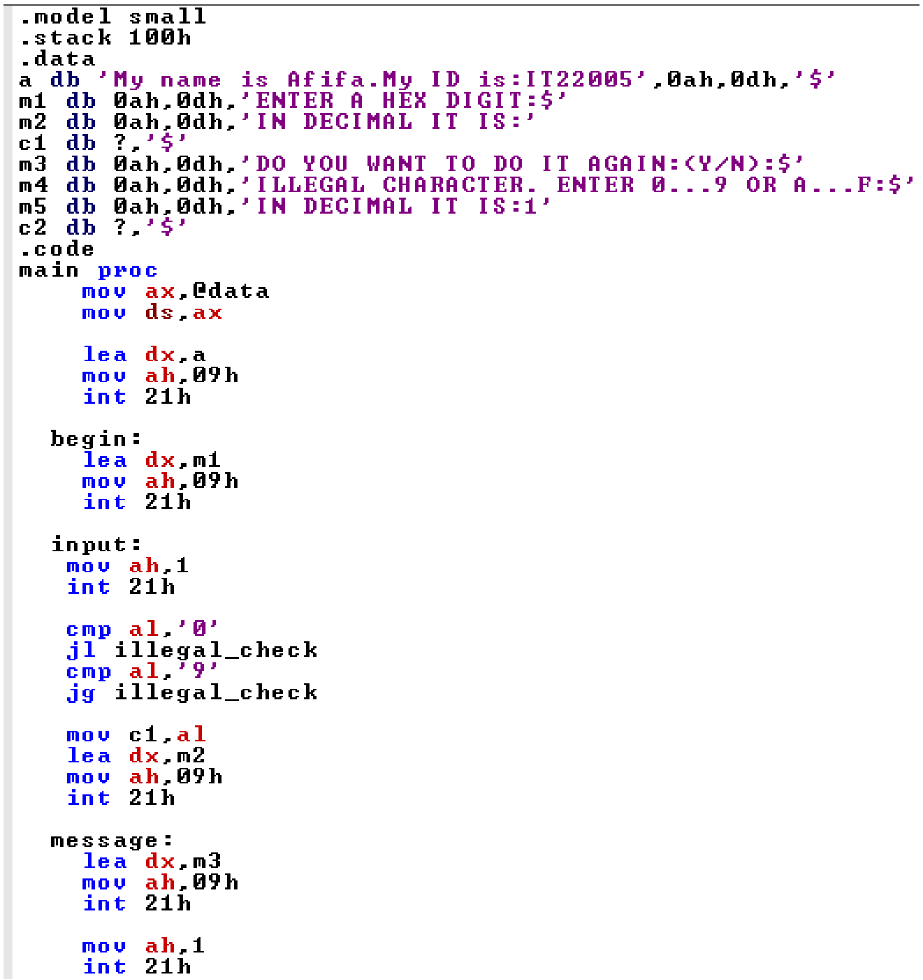
ENTER A HEX DIGIT: c

ILLEGAL CHARACTER - ENTER 0 .. 9 OR A .. F: C

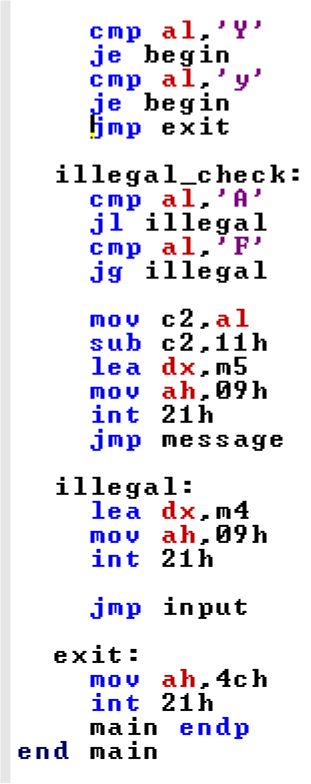
IN DECIMAL IT IS 12

DO YOU WANT TO DO IT AGAIN? N

Code:

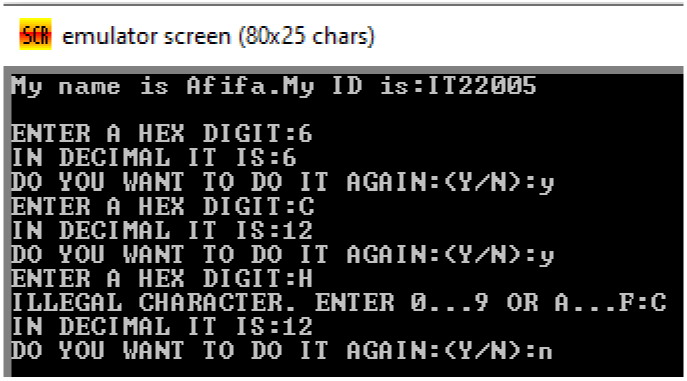


```
.model small
.stack 100h
.data
a db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'
m1 db 0ah,0dh,'ENTER A HEX DIGIT:$'
m2 db 0ah,0dh,'IN DECIMAL IT IS:'
c1 db ?,'$'
m3 db 0ah,0dh,'DO YOU WANT TO DO IT AGAIN:(Y/N):$'
m4 db 0ah,0dh,'ILLEGAL CHARACTER. ENTER 0...9 OR A...F:$'
m5 db 0ah,0dh,'IN DECIMAL IT IS:1'
c2 db ?,'$'
.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,a
    mov ah,09h
    int 21h

  begin:
    lea dx,m1
    mov ah,09h
    int 21h

  input:
   mov ah,1
   int 21h

   cmp al,'0'
   jl illegal_check
   cmp al,'9'
   jg illegal_check

   mov c1,al
   lea dx,m2
   mov ah,09h
   int 21h

  message:
    lea dx,m3
    mov ah,09h
    int 21h

    mov ah,1
    int 21h
```



```
    cmp al,'Y'
    je begin
    cmp al,'y'
    je begin
    jmp exit

  illegal_check:
    cmp al,'A'
    jl illegal
    cmp al,'F'
    jg illegal

    mov c2,al
    sub c2,11h
    lea dx,m5
    mov ah,09h
    int 21h
    jmp message

  illegal:
    lea dx,m4
    mov ah,09h
    int 21h

    jmp input

  exit:
    mov ah,4ch
    int 21h
    main endp
end main
```

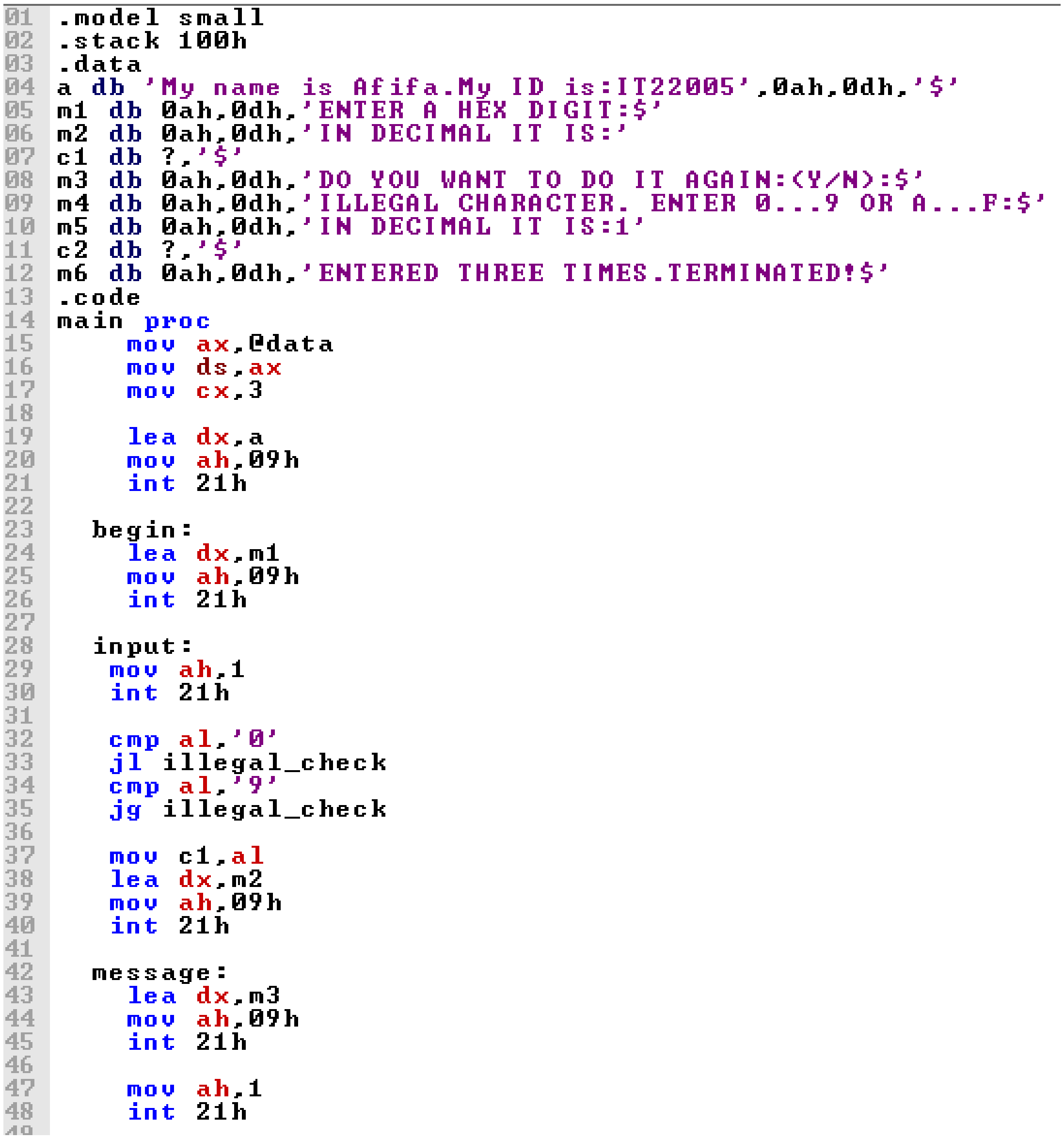
Output:



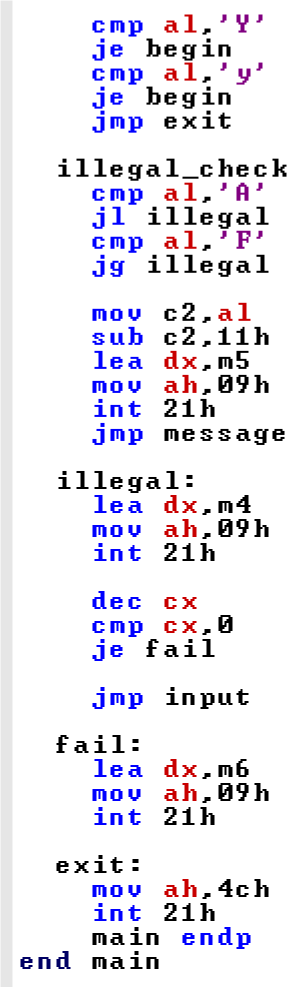
```
My name is Afifa.My ID is:IT22005

ENTER A HEX DIGIT:6
IN DECIMAL IT IS:6
DO YOU WANT TO DO IT AGAIN:(Y/N):y
ENTER A HEX DIGIT:C
IN DECIMAL IT IS:12
DO YOU WANT TO DO IT AGAIN:(Y/N):y
ENTER A HEX DIGIT:H
ILLEGAL CHARACTER. ENTER 0...9 OR A...F:C
IN DECIMAL IT IS:12
DO YOU WANT TO DO IT AGAIN:(Y/N):n
```

Program 4: A programming exercise 3, except that if the user fails to enter a hex-digit character in three tries, display a message and terminate the program.

Code:

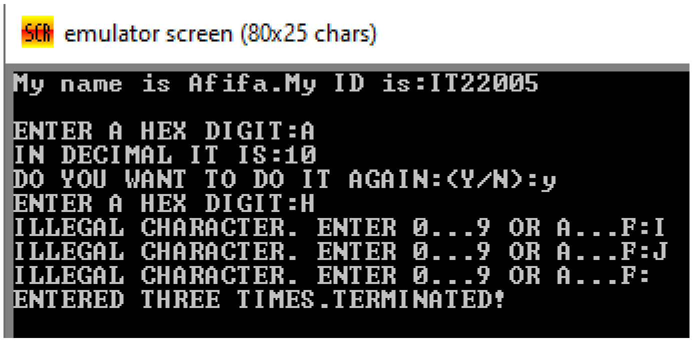


```
.model small
.stack 100h
.data
a db 'My name is Afifa.My ID is:IT22005',0ah,0dh,'$'
m1 db 0ah,0dh,'ENTER A HEX DIGIT:$'
m2 db 0ah,0dh,'IN DECIMAL IT IS:'
c1 db ?,'$'
m3 db 0ah,0dh,'DO YOU WANT TO DO IT AGAIN:(Y/N):$'
m4 db 0ah,0dh,'ILLEGAL CHARACTER. ENTER 0...9 OR A...F:$'
m5 db 0ah,0dh,'IN DECIMAL IT IS:1'
c2 db ?,'$'
m6 db 0ah,0dh,'ENTERED THREE TIMES.TERMINATED!$'
.code
main proc
    mov ax,@data
    mov ds,ax
    mov cx,3

    lea dx,a
    mov ah,09h
    int 21h

  begin:
    lea dx,m1
    mov ah,09h
    int 21h

  input:
   mov ah,1
   int 21h

   cmp al,'0'
   jl illegal_check
   cmp al,'9'
   jg illegal_check

   mov c1,al
   lea dx,m2
   mov ah,09h
   int 21h

  message:
    lea dx,m3
    mov ah,09h
    int 21h

    mov ah,1
    int 21h
```



```
      cmp al,'Y'
      je begin
      cmp al,'y'
      je begin
      jmp exit

    illegal_check:
      cmp al,'A'
      jl illegal
      cmp al,'F'
      jg illegal

      mov c2,al
      sub c2,11h
      lea dx,m5
      mov ah,09h
      int 21h
      jmp message

    illegal:
      lea dx,m4
      mov ah,09h
      int 21h

      dec cx
      cmp cx,0
      je fail

      jmp input

    fail:
      lea dx,m6
      mov ah,09h
      int 21h

    exit:
      mov ah,4ch
      int 21h
      main endp
end main
```

Output:



```
My name is Afifa.My ID is:IT22005

ENTER A HEX DIGIT:A
IN DECIMAL IT IS:10
DO YOU WANT TO DO IT AGAIN:(Y/N):y
ENTER A HEX DIGIT:H
ILLEGAL CHARACTER. ENTER 0...9 OR A...F:I
ILLEGAL CHARACTER. ENTER 0...9 OR A...F:J
ILLEGAL CHARACTER. ENTER 0...9 OR A...F:
ENTERED THREE TIMES.TERMINATED!
```