

```

print macro msg
    mov dx,offset msg
    mov ah,09h
    int 21h
endm

data segment
    p dw 00
    str1 db 0ah,0dh,"Enter string: $"
    str2 db 0ah,0dh,"Enter substring: $"
    str3 db 0ah,0dh,"Given substring is present $"
    str4 db 0ah,0dh,"Given substring is not present$"
    msg db 0ah,0dh,"Press 1 to continue & 0 to exit $"
    str5 db 20 dup(?)
    str6 db 10 dup(?)
data ends

```

```

code segment
assume cs:code,ds:data

```

```

start:

```

```

    mov ax,data
    mov ds,ax

```

```

loo:

```

```

    print str1
    mov cx,00
    mov si,offset str5

```

```

a1: mov ah,01h

```

```

    int 21h
    cmp al,13
    jz a2
    mov [si],al
    inc si
    inc cx
    jmp a1

```

```

a2: mov bl,'$'

```

```

    mov [si],bl
    print str2
    mov di,offset str6

```

```

b1: mov ah,01h

```

```

    int 21h
    cmp al,13
    jz b2
    mov [di],al
    inc di
    jmp b1

```

```

b2: mov bl,'$'

```

```

    mov [di],bl
    mov si,offset str5
    mov dh,00h

```

```

c1: cmp p,01

```

```

    je c6
    mov di,offset str6

```

```

c2: mov dl,[si]

```

```

    mov al,[di]
    cmp al,dl
    jnz c3
    inc si
    inc di
    mov p,01
    mov bl,[di]

```

```

    cmp bl,'$'
    jz c1
    loop c2
    mov p,00
    jmp c5
c3: mov p,00
    inc dh
    mov bh,dh
    mov si,offset str5
c4: cmp bh,00
    je c1
    inc si
    mov bl,[si]
    cmp bl,'$'
    jz c7
    dec bh
    jmp c4
    inc si
    jmp c2
c5: cmp p,00
    jz c7
c6: print str3
    jmp c8
    c7: print str4
    jmp c8
c8: print msg
    mov ah,01h
    int 21h
    cmp al,31h
    jnz exit
    jmp loo
exit:
    mov ah,4ch
    int 21h
code ends
end start

```

Output

```

Enter string:Hello World
Enter substring :hai
Given substring is not present
Press 1 to continue & 0 to exit 1

```

```

Enter string:Hello World
Enter substring :World
Given substring is present
Press 1 to continue & 0 to exit 0

```