```
.MODEL SMALL

newline macro
    mov ah,2
    mov dl,10
    int 21h
    mov dl,13
    int 21h


    endm

.STACK 100H

.DATA

;welcome page

data1 db 10,13

data2 db 10,13
data3 db 10,13
data4 db 10,13
data5 db 10,13
data6 db 10,13

;choose
data7 db 10,13
data8 db 10,13
data19 db 10,13
data34 db 10,13
data35 db 10,13
data36 db 10,13
data37 db 10,13
data38 db 10,13
data39 db 10,13

;booklist
data9 db 10,13
data10 db 10,13
data11 db 10,13

;english novel list
data12 db '1.Book: THINGS FALL APART : FICTION, Price: 20'
```

```
data13 db '2. Book: WUTHERING HEIGHTS, Category: Fiction, Price: 30'


;BANGLA novels list

data21 db '3.ROKTAKTO PRANTOR : FICTION, Price: 40'
data22 db '4.Book: RUDALI : FICTION, Price: 45'

;HISTORIC books list

data27 db '4.THE DIARY OF A YOUNG GIRL:HISTORY, Price: 25'
data28 db '5.MAHATMA GANDHI : HISTORY, Price: 35'
data29 db '6.THE GUNS OF AUGUST : HISTORY, Price: 15'
data30 db '7.THE DEVIL IN THE WHITE CITY : HISTORY, Price: 25'
data31 db '8.A STUDY OF HISTORY : HISTORY, Price: 45'
data32 db '9.THE MAKING OF ENGLISH WORKING CLASS : HISTORY, Price: 55'
data33 db '10.ORIENTALISM : HISTORY, Price: 15'

;DEFINE BOOK PRICES
PRICE1 DW 20
PRICE2 DW 30
PRICE3 DW 40
PRICE4 DW 25
PRICE5 DW 35
PRICE6 DW 15
PRICE7 DW 25
PRICE8 DW 45
PRICE9 DW 55
PRICE10 DW 15
total1 db 'total price:$',

; declare variables here

.CODE
MAIN PROC

; initialize DS

MOV AX,@DATA
MOV DS,AX

; enter your code here
```

```
;USE NEWLINE MACRO NEWLINE
newline
newline
;welcome page
mov ah,9
lea dx,data1
int 21h

lea dx,data2
int 21h

lea dx,data3
int 21h


lea dx,data4
int 21h

lea dx,data5
int 21h
;
lea dx,data6
int 21h

;new line
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h


; input  start

mov ah,9
lea dx, data19
int 21h

mov ah,1
int 21h
mov bh,al
sub bh,48
```

```
cmp bh,1
je booklist
jmp invalid



booklist:
;newline
mov ah,2
mov dl,10
int 21
mov dl,13
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

mov ah,9
lea dx, data7
int 21h

lea dx, data9
int 21h
lea dx, data11
int 21h


;list choose
lea dx, data8
int 21h

mov ah,1
int 21h
mov bh,al
sub bh,48

cmp bh,1
je englishnovels

cmp bh,2
```

```
je BANGLAnovels


cmp bh,3
je HISTORICbooks


jmp invalid

;english novels list
englishnovels:
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h



;start


mov ah,9
lea dx, data12
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h



mov ah,9
lea dx, data13
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
```

```
 ; user selection

 mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,1
 je take_price1
 jmp invalid



take_price1:
push price1 ;pushing into stack


jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,2
 je take_price2
 jmp invalid


take_price2:
push price2
jmp booklist




;condition checking

lea dx, data34
int 21h

mov ah,1
int 21h
mov bl,al
sub bl,48
```

```
jmp invalid


BANGLAnovels:
;newline
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h




; start

mov ah,9
lea dx,data28
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h




mov ah,9
lea dx, data21
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
```

```
mov ah,9
lea dx, data22
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
; user selection

 mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,3
 je take_price3
 jmp invalid



take_price3:
push price3 ;pushing into stack


jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,4
 je take_price4
 jmp invalid


take_price4:
push price4
jmp booklist
```

```
;condition checking;
lea dx, data34
int 21h

mov ah,1
int 21h
mov bl,al
sub bl,48



jmp invalid



;for exit
mov ah,2
mov dl,0
int 21h
mov dl,13
int 21h


mov ah,9
lea dx,data38
int 21h

mov ah,1
int 21h
mov bh,al

cmp bh,1
jmp exit



HISTORICbooks:

mov ah,2
mov dl,10
```

```
int 21h
mov dl,13
int 21h

; start
mov ah,9
lea dx, data27

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

mov ah,9
lea dx, data28
int 21h

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
lea dx, data29
int 21h


mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
lea dx,data30
int 21h



mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
lea dx, data31
int 21h
```

```
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
lea dx, data32
int 21h


mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h
lea dx, data33
int 21h


;user selection
mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,5
 je take_price5
 jmp invalid




take_price5:
push price5 ;pushing into stack


jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,6
```

```
 je take_price6
 jmp invalid


take_price6:
push price6
jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,7
 je take_price7
 jmp invalid




take_price7:
push price7 ;pushing into stack


jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,8
 je take_price8
 jmp invalid


take_price8:
push price8
jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,9
```

```
 je take_price9
 jmp invalid



take_price9:
push price9 ;pushing into stack


jmp booklist

mov ah,1
 int 21h
 mov bl,al
 sub bl,48
 cmp bl,10
 je take_price10
 jmp invalid


take_price10:
push price10
jmp booklist


;condition checking
lea dx,data34
int 21h


mov ah,1
int 21h
mov bl,al
sub bl,48




mov cx,ax
add ch,48
```

```
add cl,48


lea dx, data37
mov ah,9
int 21h


mov ah,2
mov dl,ch
int 21h


mov dl,cl
int 21h


mov dl,cl
int 21h

mov dl,48
int 21h

mov dl,47
int 21h
mov dl,45
int 21h




mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h


mov ah,9
lea dx, data38
int 21h

mov ah,9
lea dx, data39
```

```
int 21h



mov ah,9
lea dx, data8
int 21h

mov ah,1
int 21h
sub al,48

cmp al,1
je booklist

cmp al,2
je exit

jmp invalid



mov cx,ax
add ch,48
add cl,48

lea dx, data37
mov ah,9
int 21h

mov ah,2
mov dl,ch
int 21h

mov dl,cl
int 21h


mov dl,48
int 21h
```

```
mov dl,47
int 21h
mov dl,45
int 21h



mov ah,2
mov dl,10
int 21h
mov  dl,13
int 21h


mov ah,9
lea dx, data38
int 21h


mov ah,9
lea dx,data39
int 21h


mov ah,9
lea dx, data8
int 21h

mov ah,1
int 21h
sub al,48

cmp al,1
je booklist

cmp al,2
je exit
jmp invalid



mov bl,20
lea dx, data35
mov ah,9
```

```
int 21h

mov ah,1
int 21h
sub al,48

mul bl

mov cx,ax
add ch,48
add cl,48


lea dx, data37
mov ah,9
int 21h

mov ah,2
mov dl,ch
int 21h

mov dl,cl
int 21h

mov dl,47
int 21h
mov dl,45
int 21h


; exit
mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h

mov ah,9
lea dx, data38
int 21h

mov ah,9
lea dx, data39
int 21h
```

```
mov ah,9
lea dx, data8
int 21h


mov ah,1
int 21h
sub al,48



cmp al,1
je booklist

cmp al,2
je exit

jmp invalid


invalid:

mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h


mov ah,2
mov dl,10
int 21h
mov dl,13
int 21h


mov ah,1
lea dx,data36
int 21h
jmp  exit
```

```
exit:

call total_cost



;exit to DOS

MOV AX,4C00H
INT 21H

MAIN ENDP



total_cost proc
    mov ax,0;


    count_loop:
    pop cx
    add ax,cx
    cmp sp,100h
    jne count_loop




    lea dx,total1
    mov ah,9
    int 21h


    call full_count
    total_cost endp



    full_count PROC

    mov cx, 10
    MOV dx, 0
```

```
count2:
    div cx
    add dl, 48
    push dx
    mov dx, 0
    cmp ax, 0
    jne count2

value:
    pop dx
    mov ah, 2
    int 21h
    cmp sp, 100H
    jne value
    ret
full_count ENDP



    END MAIN
```