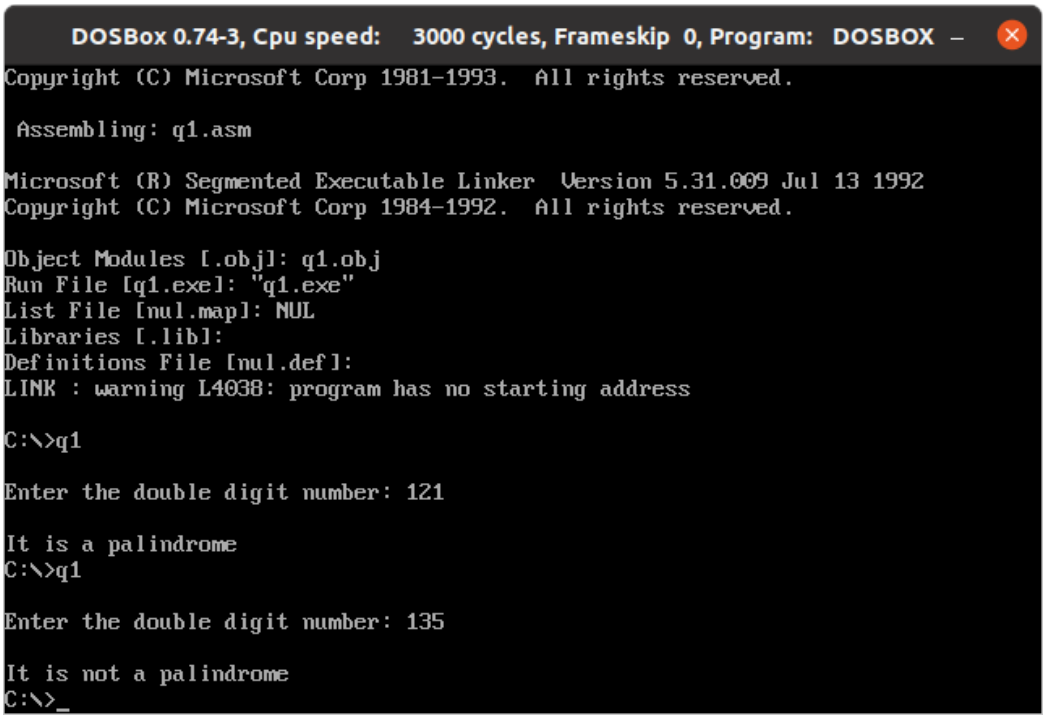
TASK 1:

```
.model small
.stack 100h
.data
recov dw 0
msg1 db 10,13,'Enter the double digit number: $'
safe dw 0
msg2 db 10,13,'It is a palindrome $'
msg3 db 10,13,'It is not a palindrome $'
rem dw 0
qou dw 0
tot dw 0
digitCount db 0
temp dw 0
num dw 0
conti dw 0
.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0



MDIS:                                   ;Input
mov dx,0
lea dx, msg1
mov ah,09h
int 21h

mov dx,0
Input:
mov ah,01
int 21H
cmp al,13
JE re
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,num
mov bl,10
```

```
mul bl
add ax,temp
mov num,ax
inc digitCount
jmp Input


re:
mov ax,num
mov safe,ax



result:
mov dx,0
mov ax,num

mov bx,0
mov bx,10
div bx
mov qou,ax
add rem,dx
mov ax,rem
mov bl,10
mul bl
mov rem,ax
mov ax,qou
mov num,ax
cmp num,0
jne result
mov dx,0
mov ax,rem
mov bx,10
div bx
mov rem,ax
mov dx,safe
mov ax,rem
cmp ax,dx
je palin
mov dx,0
lea dx, msg3
mov ah,09h
int 21h
```

```
jmp exit

palin:
mov dx,0
lea dx, msg2
mov ah,09h
int 21h




exit:
mov ah,04ch
int 21h
end
```



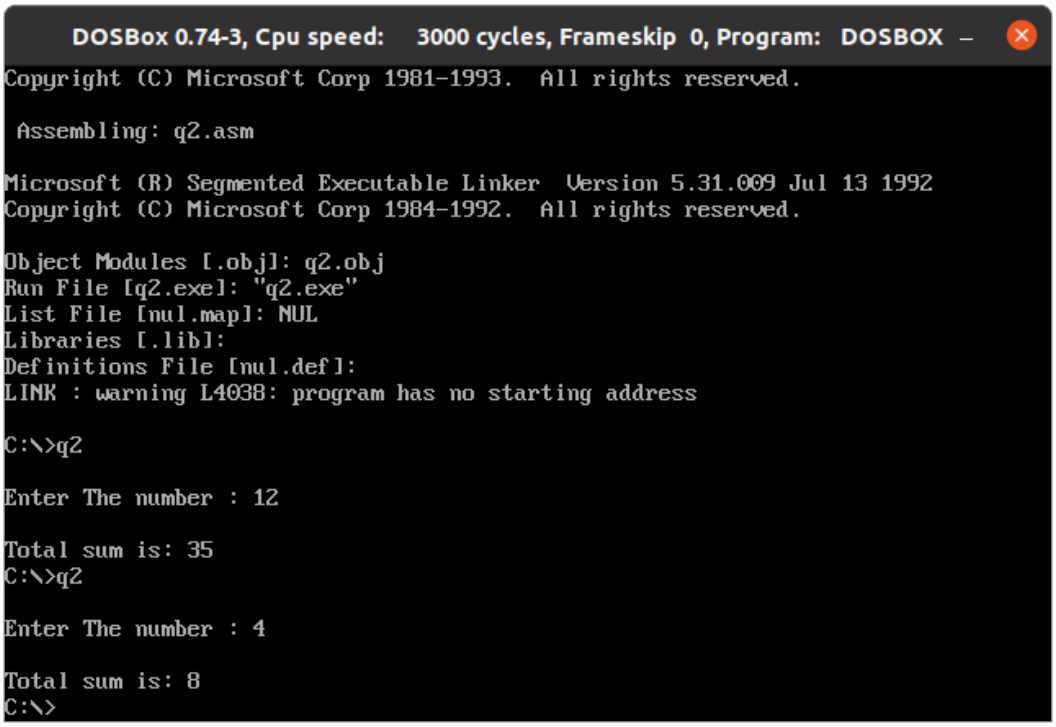
Task 2:

```
.model small
.stack 100h
.data


msg1 db 10,13,'Enter The number : $'

msg2 db 10,13,'Total sum is: $'
```

```
divi db 1
num1 dw 0

tot dw 0
digitCount db 0
temp dw 0
digitCount1 db 0

.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0


                                        ;input
mov dx,0
lea dx, msg1
mov ah,09h
int 21h


Input:
mov ah,01
int 21H
cmp al,13
JE continue
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,num1
mov bl,10
mul bl
add ax,temp
mov num1,ax
inc digitCount
jmp Input


continue:
mov bx,1
mov cx,num1
```

```
mov ax,num1
mov tot,0

result:
mov dx,0
mov ax,num1
div bx
add tot,ax
inc bx
cmp cx,num1
loop result



ans:
mov dx,0



lea dx, msg2
mov ah,09h
int 21h



val:
mov ax,tot
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov tot,ax
inc cx
cmp tot,0
jne val
display:
pop dx
add dl,48
mov ah,02
int 21h
```

```
loop display

exit:
mov ah,04ch
int 21h
End
```



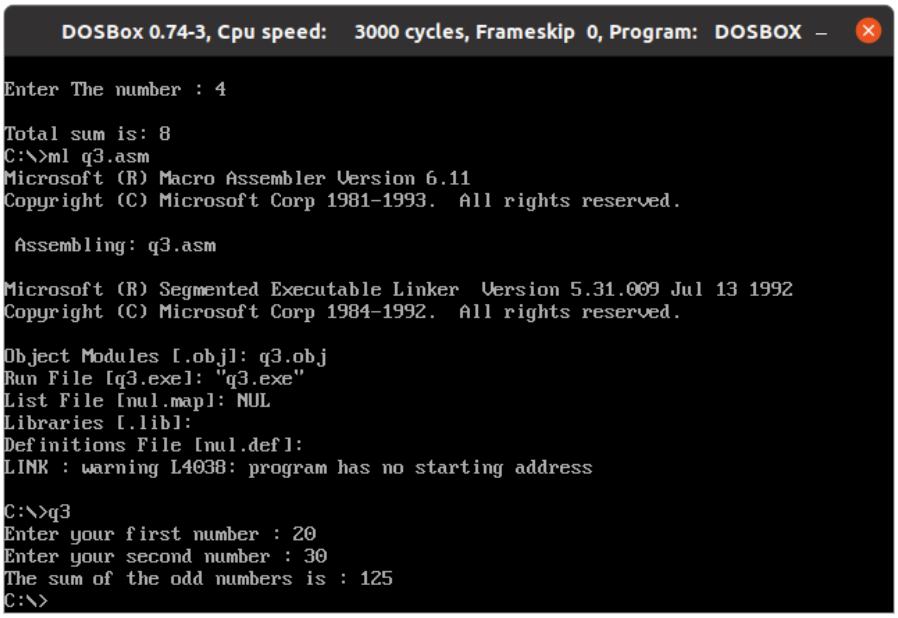
Task 3:

```
.model small
.stack 100h
.data
msg1 db 'Enter your first number : $'
msg2 db 'Enter your second number : $'
msg5 db 'The sum of the odd numbers is : $'
Number1 dw 0
Number2 dw 0
counter1 db 0
temp1 dw 0
sumodd dw 0
.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0
mov dx,0
```

```
lea dx, msg1
mov ah,09h
int 21h
Input1:
mov ah,01
int 21H
cmp al,13
JE break1
sub al,48

mov ah,0
mov temp1,ax
mov ax,0
mov ax,Number1
mov bl,10
mul bl
add ax,temp1
mov Number1,ax
jmp Input1

break1:
lea dx, msg2
mov ah,09h
int 21h
mov temp1,0
mov ax,0
mov bx,0
mov cx,0
mov dx,0
Input2:
mov ah,01
int 21H
cmp al,13
JE checkodd
sub al,48
mov ah,0
mov temp1,ax
mov ax,0
mov ax,Number2
mov bl,10
mul bl
add ax,temp1
mov Number2,ax
jmp Input2
```

```
checkodd:
mov counter1,0
mov bx,Number1
mov temp1, bx
mov ax , Number1

add Number1,1
mov bl , 2
div bl
mov al, ah
cmp al , 1
je sumcal
mov ax,Number1
cmp ax, Number2
jbe checkodd
jmp displaysum

sumcal:
mov bx , temp1
add sumodd, bx
pushodd:
mov bx,0
mov ax,temp1
mov bl ,10
div bl
mov dl,ah
push dx
mov ah,0
mov temp1,ax
inc counter1
cmp temp1,0
jne pushodd
popnum:
pop Dx
Add Dl,48
Mov Ah,02
Int 21H
Cmp counter1,1
je checkodd
Dec counter1
Jmp popnum

displaysum:
mov dl ,13
```

```
mov ah,02h
int 21h
lea dx, msg5
mov ah,09h
int 21h
mov bx, sumodd
mov temp1 ,bx
pushsum:
mov bx,0
mov ax,temp1
mov bl ,10
div bl
mov dl,ah
push dx
mov ah,0
mov temp1,ax
inc counter1
cmp temp1,0
jne pushsum
popsum:
Pop Dx
Add Dl,48
Mov Ah,02
Int 21H
Cmp counter1,1
JE exit
Dec counter1
Jmp popsum
exit:
mov ah,4ch
int 21h
end
```



Task 4:

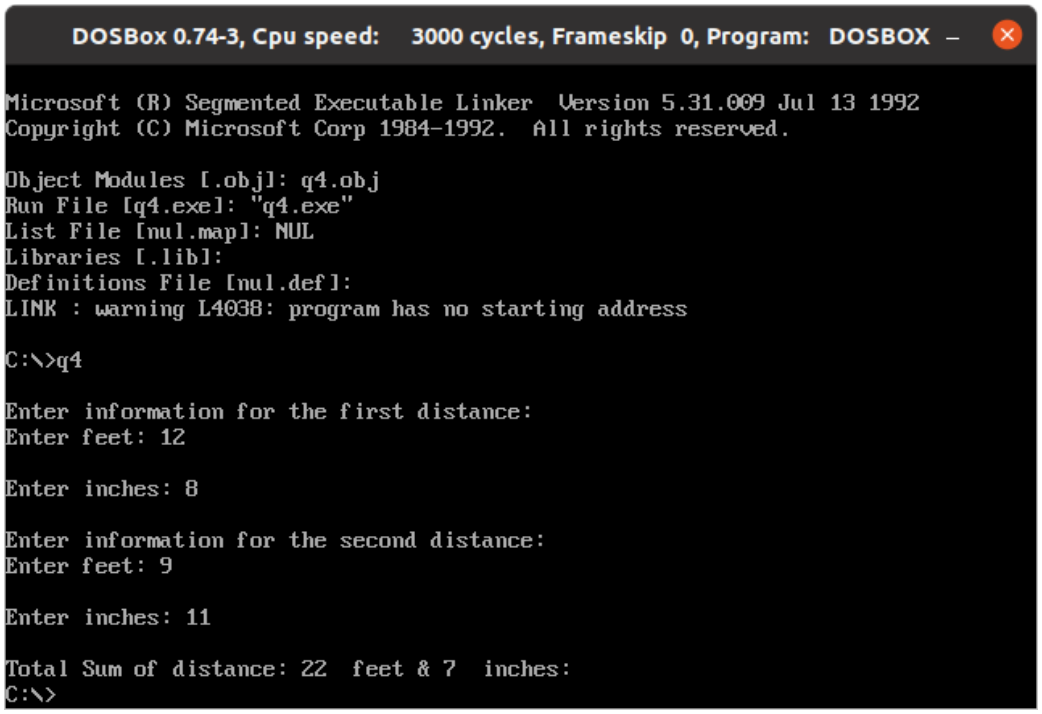
```
.model small
.stack 100h
.data
msg1 db  10,13,'Enter information for the first distance: $'
msg2 db  10,13,'Enter information for the second distance: $'
msgf db  10,13,'Enter feet: $'
msgi db  10,13,'Enter inches: $'

msgs db 10,13,'Total Sum of distance: $'
msga db '  feet & $'
msgb db  '  inches: $'
tf dw 0
ti dw 0
feet1 dw 0
inch1 dw 0

inch2 dw 0
feet2 dw 0
per dw 0
tot dw 0
temp dw 0
digitcount db 0


.code
mov ax,@data
mov ds,ax
```

```
mov ax,0
mov bx,0
mov cx,0


                                        ;input for the first dist
mov dx,0
lea dx, msg1
mov ah,09h
int 21h
lea dx, msgf
mov ah,09h
int 21h
mov dx,0
Input1:
mov ah,01
int 21H
cmp al,13
JE continue1
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,feet1
mov bl,10
mul bl
add ax,temp
mov feet1,ax
inc digitCount
jmp Input1

continue1:
lea dx, msgi
mov ah,09h
int 21h
Input2:
mov ah,01
int 21H
cmp al,13
JE continue2
sub al,48
mov ah,0
mov temp,ax
mov ax,0
```

```
mov ax,inch1
mov bl,10
mul bl
add ax,temp
mov inch1,ax
inc digitCount
jmp Input2

continue2:
mov dx,0
lea dx, msg2
mov ah,09h
int 21h
lea dx, msgf
mov ah,09h
int 21h
mov dx,0
Input3:
mov ah,01
int 21H
cmp al,13
JE continue3
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,feet2
mov bl,10
mul bl
add ax,temp
mov feet2,ax
inc digitCount
jmp Input3

continue3:
lea dx, msgi
mov ah,09h
int 21h
Input4:
mov ah,01
int 21H
cmp al,13
JE continue4
sub al,48
```

```
mov ah,0
mov temp,ax
mov ax,0
mov ax,inch2
mov bl,10
mul bl
add ax,temp
mov inch2,ax
inc digitCount
jmp Input4

continue4:
mov ax,inch1
add ax,inch2
mov ti,ax
cmp ax,12
jge addi
jmp naddi
addi:
add feet1,1
sub ax,12
mov ti,ax
naddi:
mov ax,feet1
add ax,feet2
mov tf,ax

mov dx,0
lea dx, msgs
mov ah,09h
int 21h
val:
mov ax,tf
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov tf,ax
inc cx
cmp tf,0
jne val
display:
```

```
pop dx
add dl,48
mov ah,02
int 21h
loop display

mov dx,0
lea dx, msga
mov ah,09h
int 21h
pval:
mov ax,ti
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov ti,ax
inc cx
cmp ti,0
jne pval
pdisplay:
pop dx
add dl,48
mov ah,02
int 21h
loop pdisplay
mov dx,0
lea dx, msgb
mov ah,09h
int 21h


exit:
mov ah,04ch
int 21h
end
```



Task 5:

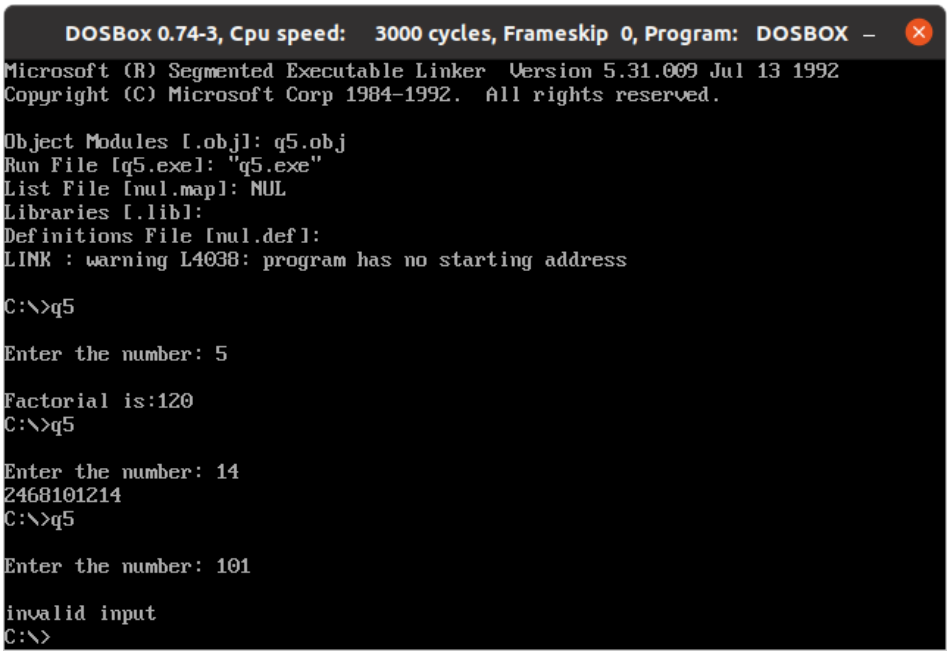
```
.model small
.stack 100h

.data
dit dw 0
s1 dw 0
s2 dw 0
msg1 db 10,13,'Enter the number: $'
msg2 db 10,13,'invalid input $'
msg3 db 10,13,'Factorial is:$'
msg4 db 10,13,'multi of 2 $'
facti dw 0
multo dw 0
temp dw 0
digitcount db 0
.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0
```

```
mov dx,0
lea dx, msg1
mov ah,09h
int 21h
mov dx,0
Input:
mov ah,01
int 21H
cmp al,13
JE continue
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,s1
mov bl,10
mul bl
add ax,temp
mov s1,ax
inc digitCount
jmp Input

continue:


mov ax,s1
mov cx,s1
mov bx,s1


cmp bx,9
jge labss

mov bx,ax
fact:
dec bx
mul bx
cmp bx,1
jne fact
mov facti,ax
```

```
mov facti,ax
mov dx,0
lea dx, msg3
mov ah,09h
int 21h
mov cx,0


pval:
mov ax,facti
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov facti,ax
inc cx
cmp facti,0
jne pval
pdisplay:
pop dx
add dl,48
mov ah,02
int 21h
loop pdisplay


jmp exit

labss:
cmp ax,99
jge labss1
mov cx,0
mov dit,0
mov multo,0
mov s2,0
multi:


add s2,2
mov ax,s2
mov multo,ax
```

```
val:
mov ax,multo
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov multo,ax
inc cx
cmp multo,0
jne val
display:
pop dx
add dl,48
mov ah,02
int 21h
loop display



add dit,2
mov cx,dit
cmp cx,s1
mov cx,0
jnge multi
jmp exit


labss1:
mov dx,0
lea dx,msg2
mov ah,09h
int 21h




exit:
mov ah,4ch
int 21h
```

end



TASK 6:

```
.model small
.stack 100h
.data
msg1 db 'Enter marks of first subject : $'
msg2 db 'Enter marks of second subject :: $'
msg3 db 'Enter marks of third subject : : $'
msg4 db 'Enter marks of fourth subject : $'
msg5 db 'Enter marks of fifth subject : $'
msg6 db 'Percentage is: $'
msg7 db 'Marks obtained out of 500 are: $'
digitCount db 0
temp dw 0
sEven dw 0
digitCount1 db 0
s1 dw 0
s2 dw 0
s3 dw 0
s4 dw 0
s5 dw 0
per dw 0

tot dw 0
```

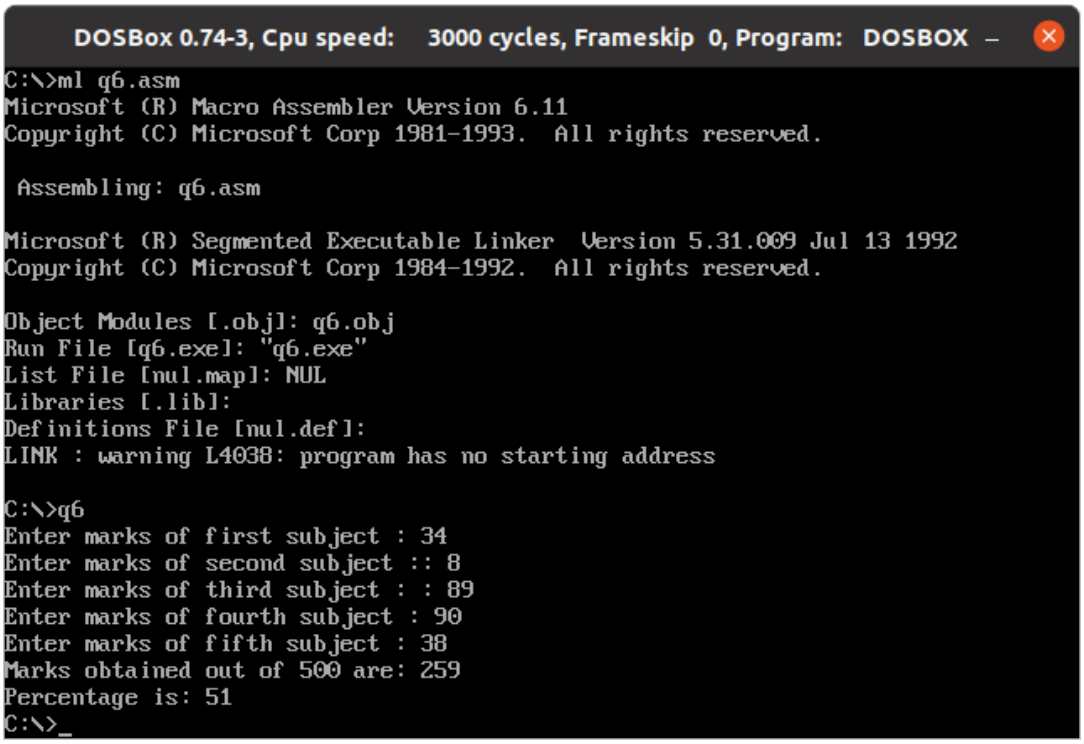
```
.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0
;input for the first subject
mov dx,0
lea dx, msg1
mov ah,09h
int 21h
mov dx,0
lnput:
mov ah,01
int 21H
cmp al,13
JE continue
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,s1
mov bl,10
mul bl
add ax,temp
mov s1,ax
inc digitCount
jmp lnput

;input for the 2nd subject

continue:
mov dx,0
lea dx, msg2
mov ah,09h
int 21h
lnput1:
mov ax,0
mov temp,ax
mov ah,01
int 21H
cmp al,13
JE continue1
```

```
sub al,48
mov ah,0

mov temp,ax
mov ax,0
mov ax,s2
mov bl,10
mul bl
add ax,temp
mov s2,ax
inc digitCount1
jmp Input1

;input for the 3rdsubject

continue1:
mov dx, offset msg3
mov ah, 09h
int 21h
Input2:
mov ax,0
mov temp,ax
mov ah,01
int 21H
cmp al,13
JE continue2
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,s3
mov bl,10
mul bl
add ax,temp
mov s3,ax
inc digitCount1
jmp Input2
continue2: ;input for the 4th subject
mov dx, offset msg4
mov ah, 09h
int 21h
Input3:
mov ax,0
mov temp,ax
```

```
mov ah,01
int 21H
cmp al,13
JE continue3

sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,s4
mov bl,10
mul bl
add ax,temp
mov s4,ax
inc digitCount1
jmp Input3

;input for the 5th subject

continue3:
mov dx, offset msg5
mov ah, 09h
int 21h
Input4:
mov ax,0
mov temp,ax
mov ah,01
int 21H
cmp al,13
JE marks
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,s5
mov bl,10
mul bl
add ax,temp
mov s5,ax
inc digitCount1
jmp Input4

;total marks obtained
```

```
marks:
mov dx, offset msg7
mov ah, 09h
int 21h
mov ax,0
add ax,s1
add ax,s2

add ax,s3
add ax,s4
add ax,s5
mov tot,ax
mov per,ax
tval:
mov ax,tot
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov tot,ax
inc cx
cmp tot,0
jne tval
display:
pop dx
add dl,48
mov ah,02
int 21h
loop display
MOV dl, 10
MOV ah, 02h
INT 21h
MOV dl, 13
MOV ah, 02h
INT 21h

;percentage

percentage:
mov dx, offset msg6
mov ah, 09h
int 21h
```

```
mov ax,0
mov ax,per
mov dx,0
mov bx,0
mov bx,5
div bx
mov per,ax



pval:
mov ax,per
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov per,ax
inc cx
cmp per,0
jne pval
pdisplay:
pop dx
add dl,48
mov ah,02
int 21h
loop pdisplay

exit:
mov ah,04ch
int 21h
End
```



Task 7:

```
.model small
.stack 100h
.data
recov dw 0
msg3 db 10,13,'Hours the car has travelled: $'
msg2 db 10,13,'Speed must be a positive value: $'
msg1 db 10,13,'Enter the speed of the vehicle in Mph: $'
msg4 db 10,13,'Hours must be greater than 0  $'
hour db 10,13,'Hour:: $'
dist db '  Distance:: $'
tot dw 0
digitCount db 0
temp dw 0
mph dw 0
conti dw 0
.code
mov ax,@data
mov ds,ax
mov ax,0
mov bx,0
mov cx,0
jmp MDIS
```

```
Inpu:
mov dx,0
lea dx, msg2
mov ah,09h
int 21h



MDIS:                                   ;Input for the speed
mov dx,0
lea dx, msg1
mov ah,09h
int 21h

mov dx,0
Input:
mov ah,01
int 21H
cmp al,13
JE continue
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,mph
mov bl,10
mul bl
add ax,temp
mov mph,ax
inc digitCount
jmp Input

                                        ;Input for the hours
continue:
cmp mph,0
jb Inpu
jmp intol



chk:
mov dx,0
```

```
lea dx, msg4
mov ah,09h
int 21h


intol:
mov dx,0
lea dx, msg3
mov ah,09h
int 21h

Input1:
mov ah,01
int 21H
cmp al,13
JE continue1
sub al,48
mov ah,0
mov temp,ax
mov ax,0
mov ax,tot
mov bl,10
mul bl
add ax,temp
mov tot,ax
inc digitCount
jmp Input1

continue1:
cmp tot,0
jbe chk


Result:
mov conti,1
mov ax,mph
mov recov,ax
jmp nope

again:

mov ax,recov
mov mph,ax
sub conti,1
```

```
mov cx,conti
l1:
add mph,ax
loop l1
add conti,1

nope:
mov dx, offset hour
mov ah, 09h
int 21h



mov dx,conti
add dx,48
mov ah,02
int 21h


mov dx, offset dist
mov ah, 09h
int 21h


pval:
mov ax,mph
mov bl,10
div bl
mov dx,0
mov dl,ah
push dx
mov ah,0
mov mph,ax
inc cx
cmp mph,0
jne pval
pdisplay:
pop dx
add dl,48
mov ah,02
int 21h
loop pdisplay
```

```
add conti,1
mov dx,conti
mov ax,tot
cmp dx,ax
jle again
jmp exit


exit:
mov ah,04ch
int 21h
End
```