

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**CSE2006 - MICROPROCESSOR AND INTERFACING**

**ALEENA 20BCE0200**

**GAURANG RASTOGI 20BCE2954**

**VIVIN D 20BCI0005**

**NEHA ELSA PAUL 20BCE2451**

**EXECUTABLE CODE OF**

**INVENTORY MANAGEMENT SYSTEM USING EMU8086**

**WINTER SEMESTER 2021-22**

**SUBMITTED TO**

**FACULTY : DR. RAJESH N**

**DATE : 23/04/2022**

**SLOT : L11+L12**

.MODEL SMALL

.STACK 100H

.DATA

;DECLARED STRINGS

UNAME DB 10,10,13, '\*\*\*\*\*\*LOGIN\*\*\*\*\*\*\*',10,10,13,'ENTER USERNAME: $'

PWORD DB 10,13,'ENTER PASSWORD: $'

TEXT DB 10,10,13,'\*\*\*\*\*\*INVENTORY\*\*\*\*\*\*\*$'

INTRO DB 10,10,13,'\*\*\*\*\*\*POS\*\*\*\*\*\*\*$'

ENTER DB 10,13,'PLEASE ENTER THE KEY OF THE ITEM YOU WANT TO BUY: $'

ENTER1 DB 10,13, 'PLEASE ENTER THE KEY OF THE ITEM WHOSE STOCKS YOU WANT TO EDIT: $'

INFO DB 10,13,' KEYS ITEMS PRICE STOCKS$'

WATERMELON DB 10, ' 1 WATERMELON JUICE RS 30 $'

PINEAPPLE DB 10,13,' 2 PINEAPPLE JUICE RS 40 $'

APPLE DB 10,13,' 3 APPLE JUICE RS 35 $'

LIMESODA DB 10,13,' 4 LIME SODA RS 25 $'

GRAPE DB 10,13,' 5 GRAPE JUICE RS 40 $'

OREO DB 10,13,' 6 OREO SHAKE RS 75 $'

KITKAT DB 10,13,' 7 KITKAT SHAKE RS 85 $'

FIGNHONEY DB 10,13,' 8 FIG & HONEY RS 70 $'

DATESHAKE DB 10,13,' 9 DATE SHAKE RS 65 $'

E\_QUANTITY DB 10,13,'ENTER QUANTITY: $'

AGAIN DB 10,13,'(1.CONTINUE BILL || 2.CHECK OUT || 3.BACK TO LOGIN || 4.EXIT)', 10,13, 'ENTER CHOICE: $'

AGAIN1 DB 10,13, '(1.ADD MORE STOCKS || 2.BACK TO LOGIN || 3.EXIT): ',10,13, 'ENTER CHOICE: $'

ER\_MSG DB 10,13,'ERROR INPUT$'

ER\_MSG1 DB 10, 13, 'NOT ENOUGH STOCK$'

ER\_MSG2 DB 10, 13, 'TOO MUCH STOCK$'

CHOICE DB 10,13,'ENTER YOUR CHOICE:$'

FT DB 10,13,'TOTAL AMOUNT IS :$'

ERR DB 0DH,0AH,'WRONG INPUT! START FROM THE BEGINNING $'

ERR2 DB 0DH,0AH,'WRONG INPUT.$'

R DB 0DH,0AH,'PRESENT AMOUNT IS : $'

ERASK DB 10,13,'START FROM THE BEGINNING $'

;DECLARED VARIABLES

A DW ?

B DW ?

C DW ?

S DW 0,'$'

I1 DB 53, '$'

I2 DW 53, '$'

I3 DW 53, '$'

I4 DW 53, '$'

I5 DW 53, '$'

I6 DW 53, '$'

I7 DW 53, '$'

I8 DW 53, '$'

I9 DW 53, '$'

TEMP DW ?

CHO DB ?

USER DB 10 DUP('$')

PASS DB 25 DUP('$')

NL DB 0DH,0AH,'$' ;NEW LINE

ADMINU DB "admin", "$"

ADMINP DB "123", "$"

POSUN DB "pos", "$"

POSPW DB "456", "$"

AST DB "\*$"

.CODE

MOV AX, @DATA

MOV DS, AX

MOV ES, AX

LOGIN:

MOV AH,06H ;CLEAR SCREEN INSTRUCTION

MOV AL,00H ;NUMBER OF LINES TO SCROLL

MOV BH,07H ;DISPLAY ATTRIBUTE - COLOR

MOV CH,00D ;START ROW

MOV CL,00D ;START COL

MOV DH,25D ;END OF ROW

MOV DL,80D ;END OF COL

INT 10H

;MOVE CURSOR TO MIDDLE

MOV AH,02H ;MOVE CURSOR INSTRUCTION

MOV BH,00H ;PAGE 0

MOV DH,00D ;ROW

MOV DL,0D ;COLUMN

INT 10H

LEA DX,UNAME ;ASK FOR USERNAME

MOV AH,9

INT 21H

MOV AH, 0AH ;TAKE USERNAME INPUT

LEA DX,USER

INT 21H

LEA SI, USER

INC SI

INC SI

LEA DI, ADMINU ;COMPARE USERNAME

MOV CX,0005H

CLD

REPE CMPSB

JNZ POSCHECK

ADPASSCHECK:

LEA DX,PWORD ;ASK FOR PASSWORD

MOV AH,9

INT 21H

MOV CX,03H

LEA SI,PASS

READ:

MOV AH,07H

INT 21H

MOV BYTE PTR[SI],AL

LEA DX, AST ;DISPLAY ASTERISK

MOV AH,9

INT 21H

INC SI

DEC CX

JNZ READ

LEA SI,PASS

LEA DI, ADMINP ;COMPARE PASSWORD

MOV CX,0003H

CLD

REPE CMPSB

MOV AH,01H

INT 21H

JZ ADMIN

JNZ LOGIN

POSCHECK:

LEA SI, USER

INC SI

INC SI

LEA DI, POSUN ;COMPARE USERNAME

MOV CX,0003H

CLD

REPE CMPSB

JNZ LOGIN

POSPASSCHECK:

LEA DX,PWORD ;ASK FOR PASSWORD

MOV AH,9

INT 21H

MOV CX,03H

LEA SI,PASS

READ2:

MOV AH,07H

INT 21H

MOV BYTE PTR[SI],AL

LEA DX, AST ;DISPLAY ASTERISK

MOV AH,9

INT 21H

INC SI

DEC CX

JNZ READ2

LEA SI,PASS

LEA DI, POSPW ;COMPARE PASSWORD

MOV CX,0003H

CLD

REPE CMPSB

MOV AH,01H

INT 21H

JZ POS

JNZ LOGIN

ADMIN:

LEA DX,TEXT ;ASK FOR USERNAME

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,INFO ;PRINT INFO STRING

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,WATERMELON ;PRINT WATERMELON STRING

MOV AH,9

INT 21H

LEA DX, I1 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,PINEAPPLE ;PRINT PINEAPPLE MALE STRING

MOV AH,9

INT 21H

LEA DX, I2 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,APPLE ;PRINT APPLE STRING

MOV AH,9

INT 21H

LEA DX, I3 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,LIMESODA ;PRINT LIMESODA STRING

MOV AH,9

INT 21H

LEA DX, I4 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,GRAPE ;PRINT GRAPE STRING

MOV AH,9

INT 21H

LEA DX, I5 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,OREO ;PRINT OREO STRING

MOV AH,9

INT 21H

LEA DX, I6 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,KITKAT ;PRINT KITKAT STRING

MOV AH,9

INT 21H

LEA DX, I7 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,FIGNHONEY ;PRINT FIGNHONEY STRING

MOV AH,9

INT 21H

LEA DX, I8 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,DATESHAKE ;PRINT DATESHAKE STRING

MOV AH,9

INT 21H

LEA DX, I9 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,ENTER1 ;PRINT ENTER STRING

MOV AH,9

INT 21H

MOV AH,1 ;TAKE AN INPUT & SAVED TO AL

INT 21H

CMP AL,49 ;IF AL=1 GO TO WATERMELONB LABEL

JE WATERMELONC

CMP AL,50 ;IF AL=2 GO TO PINEAPPLEB LABEL

JE PINEAPPLEC

CMP AL,51 ;IF AL=3 GO TO APPLEB LABEL

JE APPLEC

CMP AL,52 ;IF AL=4 GO TO LIMESODAB LABEL

JE LIMESODAC

CMP AL,53 ;IF AL=5 GO TO GRAPEB LABEL

JE GRAPEC

CMP AL,54 ;IF AL=6 GO TO OREOB LABEL

JE OREOC

CMP AL,55 ;IF AL=7 GO TO KITKATB LABEL

JE KITKATC

CMP AL,56 ;IF AL=8 GO TO FIGNHONEYB LABEL

JE FIGNHONEYC

CMP AL,57 ;IF AL=9 GO TO DATESHAKEB LABEL

JE DATESHAKEC

WATERMELONC:

MOV A,30 ;PRICE OF A PRODUCT IS MOVED TO A

LEA DX, I1

MOV TEMP, DX

JMP QUANTITY1

PINEAPPLEC:

MOV A,40

LEA DX, I2

MOV TEMP, DX

JMP QUANTITY1

APPLEC:

MOV A,35

LEA DX, I3

MOV TEMP, DX

JMP QUANTITY1

LIMESODAC:

MOV A,25

LEA DX, I4

MOV TEMP, DX

JMP QUANTITY1

GRAPEC:

MOV A,40

LEA DX, I5

MOV TEMP, DX

JMP QUANTITY1

OREOC:

MOV A,75

LEA DX, I6

MOV TEMP, DX

JMP QUANTITY1

KITKATC:

MOV A,85

LEA DX, I7

MOV TEMP, DX

JMP QUANTITY1

FIGNHONEYC:

MOV A,70

LEA DX, I8

MOV TEMP, DX

JMP QUANTITY1

DATESHAKEC:

MOV A,65

LEA DX, I9

MOV TEMP, DX

JMP QUANTITY1

QUANTITY1:

LEA DX,E\_QUANTITY ;PRINT ENTER QUANTITY STRING

MOV AH,9

INT 21H

MOV AH, 01H ;INPUTING QUANTITY

INT 21H

MOV AH,00H

CMP AL,48

JL QUANTITY1

CMP AL,57

JG QUANTITY1

SUB AL,48

MOV SI, [TEMP]

MOV DX, [SI]

ADD DX, AX

CMP DL,57 ;IF DL>9, PRINT ERROR MESSAGE

JG ERRORSTOCK1

MOV [SI], DX

LEA DX, AGAIN1 ;ASKING FOR CHOICE

MOV AH, 09

INT 21H

MOV AH, 01H

INT 21H

CMP AL,49

JE ADMIN

CMP AL,50

JE LOGIN

CMP AL, 51

JE END

POS:

LEA DX,INTRO ;PRINT INTRO STRING

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

JMP BEGINTOP ;DIRECTLY GO TO BEGINTOP LABEL WHERE USER WILL GIVE INPUT

ERROR121:

LEA DX,ER\_MSG ;PRINT ERROR MESSAGE

MOV AH,9

INT 21H

;IF USER GIVES AN ERROR THEN USER WILL BE ASKED TO INPUT AGAIN

LEA DX,ERASK

MOV AH,9

INT 21H

BEGINTOP:

LEA DX,INFO ;PRINT INFO STRING

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

INT 21H

LEA DX,WATERMELON ;PRINT WATERMELON STRING

MOV AH,9

INT 21H

LEA DX, I1 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,PINEAPPLE ;PRINT PINEAPPLE MALE STRING

MOV AH,9

INT 21H

LEA DX, I2 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,APPLE ;PRINT APPLE STRING

MOV AH,9

INT 21H

LEA DX, I3 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,LIMESODA ;PRINT LIMESODA STRING

MOV AH,9

INT 21H

LEA DX, I4 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,GRAPE ;PRINT GRAPE STRING

MOV AH,9

INT 21H

LEA DX, I5 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,OREO ;PRINT OREO STRING

MOV AH,9

INT 21H

LEA DX, I6 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,KITKAT ;PRINT KITKAT STRING

MOV AH,9

INT 21H

LEA DX, I7 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,FIGNHONEY ;PRINT FIGNHONEY STRING

MOV AH,9

INT 21H

LEA DX, I8 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,DATESHAKE ;PRINT DATESHAKE STRING

MOV AH,9

INT 21H

LEA DX, I9 ;PRINT STOCK

MOV AH,9

INT 21H

LEA DX,NL ;PRINT A NEW LINE

MOV AH,9

INT 21H

LEA DX,ENTER ;PRINT ENTER STRING

MOV AH,9

INT 21H

MOV AH,1 ;TAKE AN INPUT & SAVED TO AL

INT 21H

CMP AL,49 ;IF AL=1 GO TO WATERMELONB LABEL

JE WATERMELONB

CMP AL,50 ;IF AL=2 GO TO PINEAPPLEB LABEL

JE PINEAPPLEB

CMP AL,51 ;IF AL=3 GO TO APPLEB LABEL

JE APPLEB

CMP AL,52 ;IF AL=4 GO TO LIMESODAB LABEL

JE LIMESODAB

CMP AL,53 ;IF AL=5 GO TO GRAPEB LABEL

JE GRAPEB

CMP AL,54 ;IF AL=6 GO TO OREOB LABEL

JE OREOB

CMP AL,55 ;IF AL=7 GO TO KITKATB LABEL

JE KITKATB

CMP AL,56 ;IF AL=8 GO TO FIGNHONEYB LABEL

JE FIGNHONEYB

CMP AL,57 ;IF AL=9 GO TO DATESHAKEB LABEL

JE DATESHAKEB

JMP ERROR121 ;IF WRONG KEYWORD IS PRESSED THEN THE SHOPLIST WILL SHOW AGAIN

WATERMELONB:

MOV A,30

LEA DX, I1

MOV TEMP, DX

JMP QUANTITY

PINEAPPLEB:

MOV A,40

LEA DX, I2

MOV TEMP, DX

JMP QUANTITY

APPLEB:

MOV A,35

LEA DX, I3

MOV TEMP, DX

JMP QUANTITY

LIMESODAB:

MOV A,25

LEA DX, I4

MOV TEMP, DX

JMP QUANTITY

GRAPEB:

MOV A,40

LEA DX, I5

MOV TEMP, DX

JMP QUANTITY

OREOB:

MOV A,75

LEA DX, I6

MOV TEMP, DX

JMP QUANTITY

KITKATB:

MOV A,85

LEA DX, I7

MOV TEMP, DX

JMP QUANTITY

FIGNHONEYB:

MOV A,70

LEA DX, I8

MOV TEMP, DX

JMP QUANTITY

DATESHAKEB:

MOV A,65

LEA DX, I9

MOV TEMP, DX

JMP QUANTITY

;AFTER MOVING PRICE PROGRAM WILL JUMP TO QUANTITY LABEL

QUANTITY:

LEA DX,E\_QUANTITY ;PRINT ENTER QUANTITY STRING

MOV AH,9

INT 21H

JMP MULTI ;PROGRAM WILL GO TO MULTI LABEL WHERE THE PRICE WILL BE MILTIPLIED WITH THE AMOUNT

ASK:

LEA DX,AGAIN ;PRINT AGAIN IF USER WANTS TO BUY MORE

MOV AH,9

INT 21H

MOV AH,1 ;TAKES THE INPUT OF YES OR NO

INT 21H

MOV CHO,AL

CMP AL,49 ;IF YES, THEN AGAIN GO TO SHOPLIST MENU AND BUY AGAIN

JE BEGINTOP

CMP AL,50

JGE OUTPUT2 ;IF NO, PROGRAM WILL GIVE THE TOTAL OUTPUT

LEA DX,ER\_MSG

MOV AH,9 ;IF ANY WRONG INPUT, PRINT ERROR MESSAGE AND AGAIN ASK TO BUY AGAIN

INT 21H

JMP ASK

ERROR:

LEA DX,ER\_MSG ;PRINT ERROR MESSAGE

MOV AH,9

INT 21H

JMP QUANTITY ;JUMP TO QUANTITY LABEL

ERRORSTOCK:

LEA DX,ER\_MSG1 ;PRINT ERROR MESSAGE

MOV AH,9

INT 21H

JMP BEGINTOP

ERRORSTOCK1:

LEA DX,ER\_MSG2 ;PRINT ERROR MESSAGE

MOV AH,9

INT 21H

LEA DX, AGAIN1 ;ASKING FOR CHOICE

MOV AH, 09

INT 21H

MOV AH, 01H

INT 21H

CMP AL,49

JE ADMIN

CMP AL,50

JE LOGIN

CMP AL, 51

JE END

JMP ADMIN

MULTI:

INDEC3 PROC ;INDEC3 IS FOR TAKING INPUT FOR MULTIPLY WITH THE GIVEN AMOUNT

PUSH BX ;TAKE VALUES INTO STACK

PUSH CX

PUSH DX

XOR BX,BX ;HOLDS TOTAL

XOR CX,CX ;SIGN

MOV AH,1 ;TAKE CHARACTER IN AL

INT 21H

REPEAT4:

CMP AL,48 ;IF AL<0, PRINT ERROR MESSAGE

JL ERROR

CMP AL,57 ;IF AL>9, PRINT ERRIR MESSAGE

JG ERROR

AND AX,00FH ;CONVERT TO DIGIT

PUSH AX ;SAVE ON STACK

MOV AX,10 ;GET 10

MUL BX ;AX=TOTAL \* 10

POP BX ;GET DIGIT BACK

ADD BX,AX ;TOTAL = TOTAL X 10 +DIGIT

MOV AH,1

INT 21H

CMP AL,0DH ;CARRIAGE RETURN

JNE REPEAT4 ;IF NO CARRIEGE RETURN THEN MOVE ON

MOV AX,BX ;STORE IN AX

MOV SI,[TEMP]

MOV DX,[SI]

SUB DX,AX

CMP DL,48

JL ERRORSTOCK ;IF DL<0, PRINT ERROR MESSAGE

MOV [SI],DX

JMP MUL\_

POP DX ;RESTORE REGISTERS

POP CX

POP BX

RET ;AND RETURN

INDEC3 ENDP ;END OF INDEC3

ADD\_:

;SECOND VALUE STORED IN B

MOV B,AX

XOR AX,AX ;CLEAR AX

MOV AX,B ;MOV B TO AX

ADD A,AX ;ADD A WITH AX

MOV AX,A ;MOV A TO AX

PUSH AX ;TAKE AX INTO STACK

JMP END

SUB\_:

;SECOND VALUE STORED IN B

MOV B,AX

LEA DX,R ;PRINT PRESENT AMOUNT STRING

MOV AH,9

INT 21H

XOR AX,AX ;CLEAR AX

MOV AX,B ;MOV B TO AX

PUSH AX

ADD S,AX

JMP OUTPUT

MUL\_:

;SECOND VALUE STORED IN B

MUL A ;MULTIPLY A WITH AX

PUSH AX ;TAKE AX INTO STACK

MOV A,AX

JMP SUB\_ ;JUMP TO INP1UT\_SUB

JMP OUTPUT

INPUT\_ADD:

INDEC1 PROC ;INDEC PROC1 IS FOR ADDING THE PRESENT AMOUNTS INTO TOTAL

PUSH BX ;TAKE THE VALUES IN STACK

PUSH CX

PUSH DX

BEGIN1:

XOR BX,BX ;HOLDS TOTAL

XOR CX,CX ;SIGN

MOV AH,1 ;TAKE CHARACTER IN AL

INT 21H

REPEAT2:

;IF AL<0, PRINT ERROR MESSAGE

CMP AL,48

JL ERROR

CMP AL,57 ;IF AL>9, PRINT ERROR MESSAGE

JG ERROR

AND AX,00FH ;CONVERT TO DIGIT

PUSH AX ;SAVE ON STACK

MOV AX,10 ;GET 10

MUL BX ;AX=TOTAL \* 10

POP BX ;GET DIGIT BACK

ADD BX,AX ;TOTAL = TOTAL X 10 +DIGIT

MOV AH,1 ;TAKE VALUE INTO AL

INT 21H

CMP AL,0DH ;CARRIAGE RETURN

JNE REPEAT2 ;NO KEEP GOING

MOV AX,BX ;STORE IN AX

JMP ADD\_ ;JUMP TO ADD\_ TO STORE THE TOTAL VALUE

POP DX ;RESTORE REGISTERS

POP CX

POP BX

RET ;AND RETURN

INDEC1 ENDP

OUTPUT:

;OUTDEC PROC IS FOR GIVING THE OUTPUT OF THE PRESENT AMOUNT

OUTDEC PROC

PUSH AX ;SAVE REGISTERS

PUSH BX

PUSH CX

PUSH DX

XOR CX,CX ;CX COUNTS DIGITS

MOV BX,10D ;BX HAS DIVISOR

REPEAT1:

XOR DX,DX ;PREP HIGH WORD

DIV BX ;AX = QUOTIENT, DX=REMAINDER

PUSH DX ;SAVE REMAINDER ON STACK

INC CX ;COUNT = COUNT +1

OR AX,AX ;QUOTIENT = 0?

JNE REPEAT1 ;NO, KEEP GOING

MOV AH,2 ;PRINT CHAR FUNCTION

PRINT\_LOOP:

POP DX ;DIGIT IN DL

OR DL,30H ;CONVERT TO CHAR

INT 21H ;PRINT DIGIT

LOOP PRINT\_LOOP ;LOOP UNTILL DONE

POP DX

POP CX ;RESTORE REGISTERS

POP BX

POP AX

JMP ASK

RET

OUTDEC ENDP

OUTPUT2:

LEA DX,FT ;PRINT FINAL TOTAL

MOV AH,9

INT 21H

XOR AX,AX ;CLEAR AX

MOV AX,S ;SET AX INTO 0

;OUTDEC2 IS FOR GIVING THE TOTAL OUTPUT OF THE AMOUNT

OUTDEC2 PROC

PUSH AX ;SAVE REGISTERS

PUSH BX

PUSH CX

PUSH DX

XOR CX,CX ;CX COUNTS DIGITS

MOV BX,10D ;BX HAS DIVISOR

REPEAT12:

XOR DX,DX ;PREP HIGH WORD

DIV BX ;AX = QUOTIENT, DX=REMAINDER

PUSH DX ;SAVE REMAINDER ON STACK

INC CX ;COUNT = COUNT +1

OR AX,AX ;QUOTIENT = 0?

JNE REPEAT12 ;NO, KEEP GOING

MOV AH,2 ;PRINT CHAR FUNCTION

PRINT\_LOOP2:

POP DX ;DIGIT IN DL

OR DL,30H ;CONVERT TO CHAR

INT 21H ;PRINT DIGIT

LOOP PRINT\_LOOP2 ;LOOP UNTILL DONE

POP DX

POP CX ;RESTORE REGISTERS

POP BX

POP AX

OUTDEC2 ENDP

MOV AH,01H

INT 21H

MOV AL,CHO

MOV S,0

CMP AL, 50

JE POS

CMP AL,51

JE LOGIN

CMP AL,52

JE END

END:

MOV AH, 4CH

INT 21H