|  |
| --- |
| COAL |
| COAL PROJECT |
| CALCULATOR, LOGICAL FUNCTIONS, GPA |
| BCS-4C |
| M SARMAD MUJAHID (SP14-BCS-057) & Shahbaz Ali (SP14-BCS-080) |
| **5/27/2016** |

**Code:**

TITLE MASM Template (main.asm)

; Description:

;

; Revision date:

INCLUDE Irvine32.inc

.data

;calculator data

ar byte 0dh,0ah,0

arr byte " !!!!!!!!------------ CALCULATOR ------------!!!!!!!!" ,0dh,0ah,0

arr0 byte "\*\*\*Select the operation you want to perform\*\*\* " ,0dh,0ah,0

arr1 byte "Enter 1 for Addition" ,0dh,0ah,0

arr2 byte "Enter 2 for Subtraction" ,0dh,0ah,0

arr3 byte "Enter 3 for Multiplication" ,0dh,0ah,0

arr4 byte "Enter 4 for Division" ,0dh,0ah,0

arr5 byte "Enter 5 for GPA" ,0dh,0ah,0

arr12 byte "Enter 6 for AND operation: " ,0dh,0ah,0

arr13 byte "Enter 7 for OR operation: " ,0dh,0ah,0

arr14 byte "Enter 8 for NOR operation: " ,0dh,0ah,0

arr19 byte "Enter 9 for XOR operation: " ,0dh,0ah,0

arr22 byte "Enter 10 to terminate" ,0dh,0ah,0

arr6 byte "Enter 1st number: " ,0dh,0ah,0

arr7 byte "Enter 2nd number: " ,0dh,0ah,0

arr8 byte "Enter your choice: " ,0dh,0ah,0

arr9 byte "Invalid Choice!! " ,0dh,0ah,0

arr10 byte "Your answer is: " ,0

;arr11 byte "", 0dh,0ah,0

arr16 byte "Invalid Input!!... (Input should be 0 or 1) Enter again: " ,0dh,0ah,0

arr18 byte "Qoutient: " ,0

arr17 byte "Remainder: " ,0

val word 0

arr20 byte "Enter number bettween 0 & 1: ",0dh,0ah,0

arr21 byte "Number should be from less than or equal to 9999" ,0dh,0ah,0

arr23 byte "MATH ERROR!!" ,0dh,0ah,0

; .................... GPA DATA ................................

gpa0 real4 4.0

gpa1 real4 3.5

gpa2 real4 3.3

gpa3 real4 3.0

gpa4 real4 2.5

gpa5 real4 2.3

gpa6 real4 2.0

gpa7 real4 1.5

gpa8 real4 1.3

gpa9 real4 1.0

gpa10 real4 0.0

line byte " ",0dh,0ah,0

line0 byte "Enter Marks of 6 Subjects: ",0dh,0ah,0

line1 byte "YOUR MARKS ARE: ",0dh,0ah,0

line2 byte "YOUR GPA IS: ",0dh,0ah,0

line3 byte "WRONG INPUT!! ",0dh,0ah,0

line4 byte "ENTER MARKS (should be less than or equal to 100),, ",0dh,0ah,0

sum dword ?

; ..........................................................................................

.code

main PROC

call Clrscr ;for clearing screen

choice:

mov edx, offset ar

call WriteString ;Printing new line by moving offset of ar to edx

mov edx, offset ar

call WriteString

mov edx, offset arr ;Printing menu

call WriteString

mov edx, offset arr0 ;.....

call WriteString

mov edx, offset arr1 ;.....

call WriteString

mov edx, offset arr2 ;.....

call WriteString

mov edx, offset arr3 ;.....

call WriteString

mov edx, offset arr4 ;.....

call WriteString

mov edx, offset arr5

call WriteString

mov edx, offset arr12

call WriteString

mov edx, offset arr13

call WriteString

mov edx, offset arr14 ;.......

call WriteString

mov edx, offset arr19 ; asking for xor .......

call WriteString

mov edx, offset arr22 ; asking for terminate

call WriteString

mov edx, offset arr8 ; asking for choice

call WriteString

;mov eax, options

call ReadInt ; taking choice in eax

.IF eax==1

call inputnums ; calling fuction for taking numbers from user... recieving two numbers in ebx and esi with eax=0

mov edx, offset arr10 ; printing "your answer is:"

call WriteString

add eax, ebx ;eax = 0 + 1st num

add eax, esi ;eax = 1st num + 2nd num

call WriteInt ;printing sum

jmp choice ;jump to menu

.ELSEIF eax==2

call inputnums ; calling fuction for taking numbers from user... recieving two numbers in ebx and esi with eax=0

mov edx, offset arr10 ; printing "your answer is:"

call WriteString

sub ebx, esi ;ebx = 1st num - 2nd num

mov eax, ebx ;moving answer stored in ebx to eax

call WriteInt ;print answer stored in eax

jmp choice ;jump to menu

.ELSEIF eax==3

call inputnums ; calling fuction for taking numbers from user... recieving two numbers in ebx and esi with eax=0

mov edx, offset arr10 ; printing "your answer is:"

call WriteString

mov eax, ebx ; moving 1st number to eax

mul esi ;mutiplyling 2nd number with eax and storing ans in eax

call WriteInt ;printing answer stored in eax

jmp choice ;;jump to menu

.ELSEIF eax==4

call inputnums

.IF(esi==0)

mov edx, offset arr23 ;printing "math error!!"

call WriteString

jmp choice ;restarting calculator

.endif

mov edx, offset arr18

call WriteString ; printing "Qoutient: "

mov edx, 0 ;Empties edx

mov eax, ebx ;moving dividant in eax

div esi ;eax = eax / 2nd value

call WriteInt ;printing qoutient stored in eax

mov eax, edx ;moving remainder to eax

mov edx, offset ar ; printing new line

call WriteString

mov edx, offset arr17

call WriteString ; printing Remainder that is moved in eax

call WriteInt ; printing remainder value

jmp choice ; restarting calculator

.ELSEIF eax==5 ; for gpa

call gpa ; calling gpa procedure

jmp choice ; jumping to menu or restarting calculator

.ELSEIF eax==6 ;AND OPERATION

call inputlogical ;calling inputlogical procedure for two bits values 0 or 1 in ebx and esi same as above

mov edx, offset arr10 ; printing "your ans:"

call WriteString

mov eax, ebx ;same as multiply above

mul esi

call WriteInt

jmp choice ;restarting calculator

.ELSEIF eax==7 ;Or Operation

call inputlogical

mov edx, offset arr10 ;same as and

call WriteString

.IF(ebx == 1 && esi == 1) ;if both inputs are 1 ans is one... nested if statement

mov eax, ebx

mul esi

.ELSE ;0+1 or 1+0

add eax, ebx

add eax, esi

.ENDIF

call WriteInt ;printing answer

jmp choice ;restarting calculator

.ELSEIF eax==8 ;nor operation

a:

mov edx, offset arr20 ;asking for one number b/w 0 or 1

call WriteString

call ReadInt ;input number

.if(eax==0) ;nested if

add eax,1 ;if number is zero it will become 1

.elseif(eax==1)

mul val ;if number is one it will become 0... value = 0

.else

mov edx, offset line3 ; printing wrong input

call WriteString

jmp a ; jumping to a for input again

.endif

call WriteInt ;print ans stored in eax

jmp choice ; restarting calculator

.ELSEIF eax==9 ;xor

call inputlogical

mov edx, offset arr10 ;same as and

call WriteString

.IF(ebx == esi ) ;if both inputs are 1 ans is one... nested if statement

mov eax, ebx

mul val

.ELSE ;0+1 or 1+0

add eax, ebx

add eax, esi

.ENDIF

call WriteInt ;printing answer

jmp choice ;restarting calculator

.ELSEIF eax==10 ; terminating calculator

.ELSE ; if choice is not from 1-9

mov edx, offset arr9 ; then print Invalid choice

call WriteString

jmp choice ;restarting calculator

.ENDIF ;ending IF

exit ;EXIT OF MAINPROC

;...................................

inputnums PROC ;PROCEDURE TO INPUT NUMBERS

;print "enter 1st num"

mov edx, offset arr6

call WriteString

call ReadInt

;input 1st num

mov ebx , eax ;moving 1st number to ebx

mov edx, offset arr7 ;print enter 2nd num

call WriteString

call ReadInt ;input 2nd num

mov esi, eax ;moving 2nd number to esi

mov eax, 0 ;CLEARING EAX

ret ;RETURNING BACK FROM where it is called

inputnums ENDP ;ENDING PROCEDURE

;........................................

inputlogical PROC ;procedure for input of logical functions

;print enter 1st num

mov edx, offset arr6

call WriteString

;input 1st num

prompt:

call ReadInt

.IF(eax!=0 && eax!=1) ; number should be 0 or 1

mov edx, offset arr16

call WriteString

jmp prompt

.ENDIF

mov ebx , eax

;print enter 2nd num

mov edx, offset arr7

call WriteString

;input 2nd num

prompt2:

call ReadInt

.IF(eax!=0 && eax!=1) ; number should be 0 or 1

mov edx, offset arr16

call WriteString ;printing number is invalid... enter again

jmp prompt2 ;humping above to ask again

.ENDIF

mov esi, eax

mov eax, 0 ; Empties the eax register

ret

inputlogical ENDP ;ENDING

;.............................................

gpa PROC

MOV ECX,6 ; Runs the loop 6 times

mov eax,0 ; Empties the eax register

mov sum,0 ; Move 0 to sum

mov edx, offset line0 ; Move offset of line0 to edx

call writestring ; Prints the line0 to console window

LOOP1: ; Loop to get input from user 6 times

input: ; LABEL for jump condition

call readint ; Gets the input

.if(eax>100) ; Conditional statement for user input

mov edx, offset line3

call writestring

mov edx, offset line0

call writestring

jmp input ; Jumps back to LABEL input

.endif

add sum,eax ; Adds the input to Varable sum

loop LOOP1 ; Loop Ends

mov edx, offset line1 ; Move offset of line1 to edx

call writestring ; Prints the line1 to console window

mov eax, sum ; Moves the added input back to eax register

call writeint ; Prints the added input

mov edx, offset line

call writestring ; Just for new line

; Conditional Statements to check values for generating GPA

; fld command loads the float point value to be printed

.if(eax>=540)&&(eax<=600)

mov edx, offset line2

call writestring

fld gpa0

call writefloat

.endif

.if(eax>=510)&&(eax<540)

mov edx, offset line2

call writestring

fld gpa1

call writefloat

.endif

.if(eax>=480)&&(eax<510)

mov edx, offset line2

call writestring

fld gpa2

call writefloat

.endif

.if(eax>=450)&&(eax<480)

mov edx, offset line2

call writestring

fld gpa3

call writefloat

.endif

.if(eax>=420)&&(eax<450)

mov edx, offset line2

call writestring

fld gpa4

call writefloat

.endif

.if(eax>=390)&&(eax<420)

mov edx, offset line2

call writestring

fld gpa5

call writefloat

.endif

.if(eax>=360)&&(eax<390)

mov edx, offset line2

call writestring

fld gpa6

call writefloat

.endif

.if(eax>=330)&&(eax<360)

mov edx, offset line2

call writestring

fld gpa7

call writefloat

.endif

.if(eax>=300)&&(eax<330)

mov edx, offset line2

call writestring

fld gpa8

call writefloat

.endif

.if(eax>=270)&&(eax<300)

mov edx, offset line2

call writestring

fld gpa9

call writefloat

.endif

.if(eax<270)

mov edx, offset line2

call writestring

fld gpa10

call writefloat

.endif

ret

gpa ENDP

main ENDP

END main