****

**Project COAL**

**Course: COAL**

**Submitted to:**

**Sir Abo Bakar Aslam**

**Submitted by:**

**M.Bilal**

**Roll no:**

**Roll no**

**Section:**

**BS CS**

include irvine32.inc

.data

space db " ",0

E\_ARRAY dd 10 dup(0)

O\_ARRAY dd 10 dup(0)

two dd 2

x dd 0

y dd 0

temp dd 0

choice dd 0

str1 db "Press 1 for BITWISE operations between registers.",0ah

str2 db "Press 2 for ADDITION operations between registers.",0ah

str3 db "Press 3 for SUBTRACTION operations between registers.",0ah

str4 db "Press 4 for MULTIPLICATION operations between registers.",0ah

str5 db "Press 5 for DIVISION operations between registers.",0ah

str6 db "Press 6 for operations related to LOOP, ARRAY and STACK",0ah

str7 db "Press 0 for EXIT program execution ",0ah,0

str8 db "Enter a number =",0

str9 db "Press 1 for BITWISE OR operation between registers",0ah

str10 db "Press 2 for BITWISE AND operation between registers",0ah,0

str11 db "\*\*\*\*\*\*\*\*\*\*BITWISE FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str12 db "\*\*\*\*\*\*\*\*\*\*ADDITION FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str13 db "\*\*\*\*\*\*\*\*\*\*SUBTRACTION FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str14 db "\*\*\*\*\*\*\*\*\*\*MULTIPLICATION FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str15 db "\*\*\*\*\*\*\*\*\*\*DIVISION FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str16 db "\*\*\*\*\*\*\*\*\*\*ARRAY FUNCTION CALLED\*\*\*\*\*\*\*\*\*\*",0ah,0

str17 db "Elements in E\_ARRAY ",0ah,0

str18 db "Elements in O\_ARRAY ",0ah,0

str19 db "Result Stored in EAX",0ah,0

str20 db "INVALID INPUT.....Enter again",0ah,0

.code

main proc

menu:

lea edx,str1

call writestring

call readint

mov choice,eax

.if(choice==1)

call bitwise

jmp menu

.endif

.if(choice==2)

call addition

jmp menu

.endif

.if(choice==3)

call subtraction

jmp menu

.endif

.if(choice==4)

call multiplication

jmp menu

.endif

.if(choice==5)

call division

jmp menu

.endif

.if(choice==6)

call array

jmp menu

.endif

.if(choice==0)

exit

.else

lea edx,str20

call writestring

jmp menu

.endif

main endP

BITWISE proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

call crlf

lea edx,str11

call writestring

call crlf

lea edx,str8

call writestring

call readint

mov bl,al

lea edx,str8

call writestring

call readint

mov cl,al

call crlf

label1:

lea edx,str9

call writestring

call readint

mov edx,eax

.if(edx==1)

or cl,bl

mov al,cl

.else

.if(edx==2)

and cl,bl

mov al,cl

.else

call crlf

lea edx,str20

call writestring

call crlf

jmp label1

.endif

.endif

lea edx,str19

call writestring

call dumpregs

call crlf

ret

BITWISE endp

ADDITION proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

call crlf

lea edx,str12

call writestring

call crlf

lea edx,str8

call writestring

call readint

mov bl,al

call crlf

lea edx,str8

call writestring

call readint

add al ,bl

call crlf

lea edx,str19

call writestring

call crlf

call dumpregs

ret

ADDITION endp

SUBTRACTION proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

call crlf

lea edx,str13

call writestring

call crlf

lea edx,str8

call writestring

call readint

mov bl,al

lea edx,str8

call writestring

call readint

sub bl,al

mov eax,ebx

call crlf

lea edx,str19

call writestring

call dumpregs

ret

SUBTRACTION endp

MULTIPLICATION proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

call crlf

lea edx,str14

call writestring

call crlf

lea edx,str8

call writestring

call readint

mov bl,al

lea edx,str8

call writestring

call readint

mul bl

call crlf

lea edx,str19

call writestring

call dumpregs

ret

MULTIPLICATION endp

DIVISION proc

mov eax,0

mov ebx,0

mov ecx,0

mov edx,0

call crlf

lea edx,str15

call writestring

call crlf

lea edx,str8

call writestring

call readint

mov bl,al

lea edx,str8

call writestring

call readint

mov cl,al

mov al,bl

mov edx,0

div cl

call crlf

lea edx,str19

call writestring

call dumpregs

ret

DIVISION endp

ARRAY proc

call crlf

lea edx,str16

call writestring

call crlf

mov temp,1

.while temp<=10 ;pushing elementes in stack

lea edx,str8

call writestring

call readint

push eax

inc temp

.endw

call crlf

mov temp,1

mov esi,0

mov ecx,0

.while temp<=10 ;checking even and odd and putting in arrays

mov edx,0

pop eax

mov ebx,eax

div two

.if(edx==0)

mov [E\_ARRAY+ecx],ebx

inc x

add ecx,4

.else

mov [O\_ARRAY+esi],ebx

add esi,4

inc y

.endif

inc temp

.endw

call crlf

mov ecx,0

mov temp,1

lea edx,str17

call writestring

mov esi,x

.while temp<=esi ;showing data of even array

mov eax,[E\_ARRAY+ecx]

add ecx,4

call writedec

lea edx,space

call writestring

inc temp

.endw

call crlf

call crlf

lea edx,str18

call writestring

mov ecx,0

mov temp,1

mov esi,y

.while temp<=esi ;showing data of odd array

mov eax,[O\_ARRAY+ecx]

add ecx,4

call writedec

lea edx,space

call writestring

inc temp

.endw

call crlf

ret

ARRAY endp

end