6/16/2021

**Amina i iiQadeer-359607**

**CE-42-A**

**CO-LAB i iiFINAL i ii12**

**Submitted i iito:**

**Maam i iiRimsha i iiTariq**

**Computer i iiOrganization**

**CO-Lab i iiFinal i ii12**

**i iiLAB i ii12: i iiMIPS i iiProcedures**

**Think:**

1. **What i iiis i iistack?**

In i iicomputer i iiscience, i iia i iistack i iiis i iian i iiabstract i iidata i iitype i iithat i iiserves i iias i iia i iicollection i iiof i iielements, i iiwith i iitwo i iimain i iiprincipal i iioperations: i iiPush, i iiwhich i iiadds i iian i iielement i iito i iithe i iicollection, i iiand i iiPop, i iiwhich i iiremoves i iithe i iimost i iirecently i iiadded i iielement i iithat i iiwas i iinot i iiyet i iiremoved. **i iiStacks** i iihave i iiseveral i iiapplications i iiin i iicommuter i iiprogramming. i ii

**LIFO** i ii**stacks**, i ii(**last i iiin i iifirst i iiout**) i iifor i iiexample, i iican i iibe i iiused i iito i iiretrieve i iirecently i iiused i iiobjects, i iifrom i iia i iicache. i ii**Stacks** i iimake i iiexcellent i iimechanisms i iifor i iitemporary i iistorage i iiof i iiinformation i iiwithin i iiprocedures. i ii

The i iiqueue i iidata i iistructure i iialso i iifollows i iithe i ii**FIFO** i ii(**First i iiIn i iiFirst i iiOut**) i iiprinciple, i iii.e. i iithe i iielement i iiinserted i iiat i iifirst i iiin i iithe i iilist, i iiis i iithe i iifirst i iielement i iito i iibe i iiremoved i iifrom i iithe i iilist.

1. **Describe i iidifference i iisegments i iiof i iia i iiprogram**

Every i iiinstruction i iiand i iidirective i iimust i iicorrespond i iito i ii**a i iisegment**. i iiNormally i iia i iiprogram i iiconsists i iiof i iithree i ii**segments**: i iithe i iistack, i iithe i iidata, i iiand i iithe i iicode i ii**segments**.

In i iicomputing, i iia i iicode i ii**segment**, i iialso i iiknown i iias i iia i iitext i ii**segment** i iior i iisimply i iias i iitext, i iiis i iia i iiportion i iiof i iian i iiobject i iifile i iior i iithe i iicorresponding i iisection i iiof i iithe i ii**program's** i iivirtual i iiaddress i iispace i iithat i iicontains i iiexecutable i iiinstructions. i iiEach i ii**segment** i iiis i iiused i iito i ii**contain** i iia i iispecific i iitype i iiof i iidata. i iiOne i ii**segment** i iiis i iiused i iito i ii**contain** i iiinstruction i iicodes, i iianother i ii**segment** i iistores i iithe i iidata i iielements, i iiand i iia i iithird i ii**segment** i iikeeps i iithe i ii**program** i iistack. i ii

**LAB i iiTASKS:**

**a)** i iiint i iifoo(int i iix)

i ii{

i iiIf i ii(x i ii== i ii0) i iireturn i ii0;

i iielse

i iireturn i iifoo(x i ii- i ii1);

i ii}

**Compiled i iiMIPS i iicode: i ii**

#AMINA i iiQADEER-CO i iiLAB i ii12

#Q2:

.data

i ii i ii i ii i ii i ii

.text

.globl i iimain

main: i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii

i ii i ii i ii i ii i ii#The i iiparameter i iivariables i iig, i iih, i iii, i iiand i iij i iicorrespond i iito i iithe i iiargument i iiregisters i ii$a0,$a1, i ii$a2, i ii$a3, i iiand i iif i iicorrespond i iito i ii$s0.

i ii i ii i ii i ii i ii# i iiResult i iiin i ii$v0 i ii

li i ii$a0,1 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii

li i ii$a1,2

li i ii$a2,3 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii

li i ii$a3,4 i ii i ii i ii i ii i ii

jal i iileaf\_example i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii

# i iijump-and-link i iiinstruction i iijal i iito i iicall i iifunctions i iiThe i iijal i iisaves i iithe i iireturn i iiaddress i ii i ii i ii i ii i ii i ii(the i iiaddress i iiof i iithe i iinext i iiinstruction) i iiin i iithe i iidedicated i iiregister i ii$ra, i iibefore i iijumping i iito i iithe i iifunction. i ii... i iiTo i iitransfer i iicontrol i iiback i iito i iithe i iicaller, i iithe i iifunction i iijust i iihas i iito i iijump i iito i iithe i iiaddress i iithat i iiwas i iistored i iiin i ii$ra.

add i ii$s0, i ii$v0, i ii$zero i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iireturn i iivalue(final), i iiusing i iiadd i iior i iimove, i iifrom i ii$v0 i iito i ii$s0 i iiregister

leaf\_example:

addiu i ii$sp, i ii$sp, i ii-4 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iiadjust i iistack i iito i iimake i iispace i iior i iiroom i ii i iifor i iisave i iiregisters

sw i ii$s0, i ii0($sp) i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iisave i iiregister i ii$s0 i iion i iistack i iifor i iiuse i iiafterwards

add i ii$t0, i ii$a0, i ii$a1 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iiregister i ii$t0 i iicontains i iiinstruction i ii i iig i ii+ i iih

add i ii$t1, i ii$a2, i ii$a3 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iiregister i ii$t1 i iicontains i iiinstruction i iii i ii+ i iij

sub i ii$s0, i ii$t0, i ii$t1 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iif i ii= i ii$t0 i ii- i ii$t1, i iiwhich i iiis i ii(g i ii+ i iih) i ii- i ii(i i ii+ i iij)

add i ii$v0, i ii$s0, i ii$zero i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iireturns i iif i ii($v0 i ii= i ii$s0 i ii+ i ii0)-we i iireturn i iivalues i iifrom i iifunctions

lw i ii$s0, i ii0($sp) i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iirestore i iiregister i ii$s0 i iifor i iicaller

addi i ii$sp, i ii$sp, i ii4 i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii#adjust i iistack i iito i iidelete i iisave i iiregisters i ii(load i iiword i iiin i iiorder i iito i ii"push" i iiin i iistack)

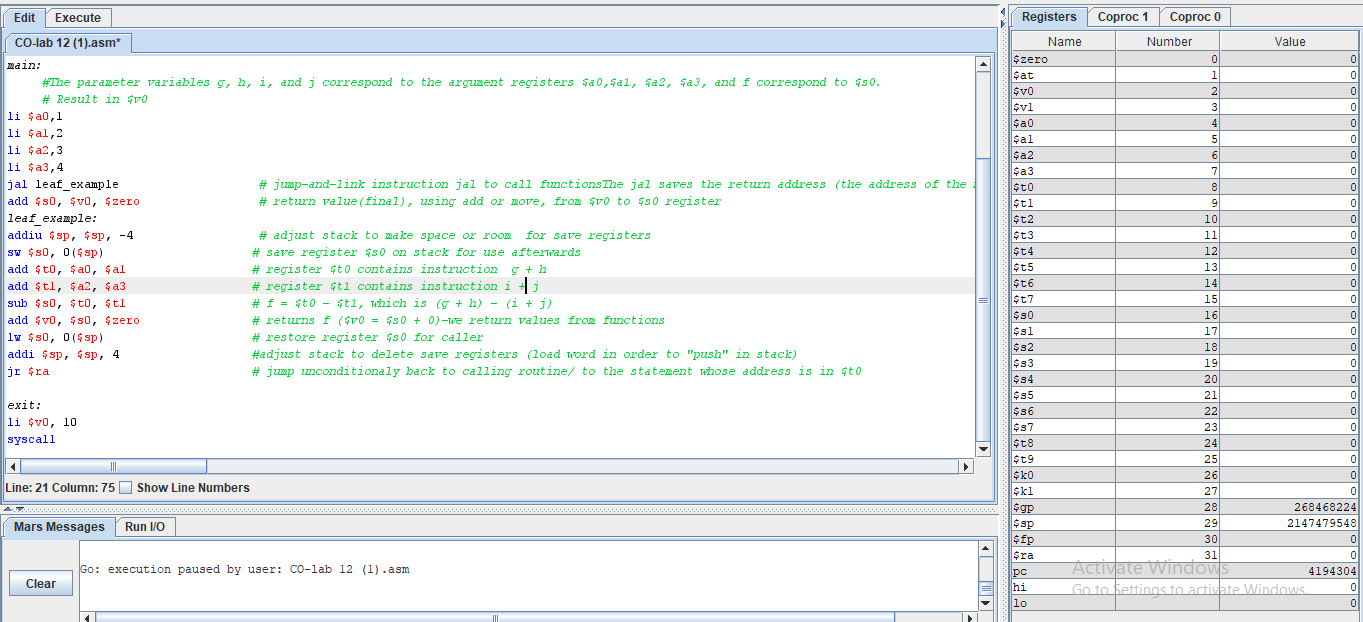
jr i ii$ra i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii# i iijump i iiunconditionally i iiback i iito i iicalling i iiroutine/ i iito i iithe i iistatement i iiwhose i iiaddress i iiis i iiin i ii$t0

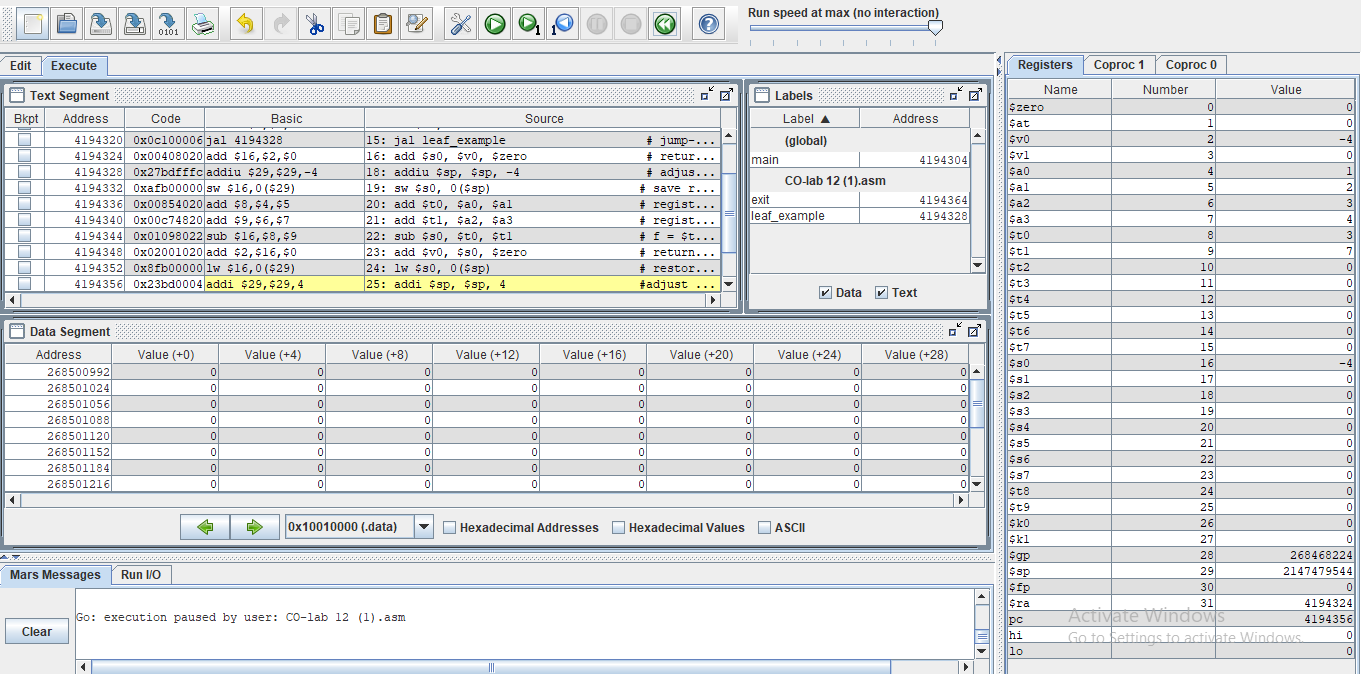
exit: i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii#defining i iiExit i iilabel i iifor i iitermination i iiof i iicode

li i ii$v0, i ii10

Syscall i ii i ii

**OUTPUT:**





**QUESTION i ii2:**

**a) i iiint i iifoo(int i iix)**

**i ii{**

**i iiif i ii(x i ii== i ii0) i iireturn i ii0;**

**i iielse**

**i iireturn i iifoo(x i ii- i ii1);**

**i ii}**

**MIPS i iiAssembly i iiCode:**

#AMINA i iiQADEER i iiCO i iiLAB i ii12

#Q2:

#AMINA iQADEER iCO iLAB i12

#Q2:

.data

.text

.globl imain

main: i

li i$a0,2 i i i i i i i i i i i i i i i i i i i i i i i i i i i i#Argument in iin i$a0,int ix=0

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i

jal ifoo i i

move i$s0, i$v0 i i

j iexit i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i

foo: i

addi i$sp, i$sp, i-8 i i i i i i i i i i i i i i i i i i i i i i# iadjust istack ifor i2 iitems i, isp iregister iis iused ito iindicate ithe ilocation iof ithe ilast iitem iput ionto ithe istack. i

sw i$ra, i4($sp) i i i i i i i i i i i i i i i i i i i i i i i i i# isave ithe ireturn iaddress ito ifourth iindex iof istack

sw i$a0, i0($sp) i i i i i i i i i i i i i i i i i i i i i i i i i# isave ithe iargument in iat ioth istack i

beqz i$a0,z i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# itest ifor ithe igiven icondition i(x==0)using ibranch iequal ito iszero iinstructon

bgtz i$a0, istep i i i i i i i i i i i i i i i i i i#applying ifor ielse icondition iusing ibranch igreater ithan izero i,then imove ito ilabel i"step"

z:

#move i$zero,$a0 i i i i i i i i i i i i i i i i i i i i i i i i i i#label iz idefined

addi i$v0, i$zero, i0 i i i i i i i i i i i i i i i i i i i i i# iif i(x==0), iresult iis i0 i

i i i i i i i i i i i i i i i i i i i i i i

jr i$ra i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# ireturn ito icaller

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#shows ithe isubroutine ireturning ito ithe ireturn iaddress ithat iwas iloaded iinto i$ra iby ithe ijal iinstruction iin ithe icaller.

step: i

addi i$a0,$a0,-1 i i i i i i i i i i i i i i i i i i i i i i i i i# ielse in i>= i1; iagrument igets i(n-1) i

jal ifoo i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# icall ifact iwith i(n-1) i

lw i$a0, i0($sp) i i i i i i i i i i i i i i i i i i i i i i i i i# ireturn ifrom ijal: irestore ioriginal in i

lw i$ra, i4($sp) i i i i i i i i i i i i i i i i i i i i i i i i i# irestore ithe ireturn iaddress i

addi i$sp, i$sp, i8 i i i i i i i i i i i i i i i i i i i i i i i# iadjust istack ipointer ito ipop i2 iitems i

move i$v0, i$a0 i i i i i i i i i i i i i i i i i i i i i i i i i i# ireturn i(n-1) i

jr i$ra i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iand ireturn ito ithe icaller i#The ijr iinstruction ireturns icontrol ito ithe icaller.

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iIt icopies ithe icontents iof i$ra iinto ithe iPC: i... iLike iall ijump iand ibranch iinstructions, i

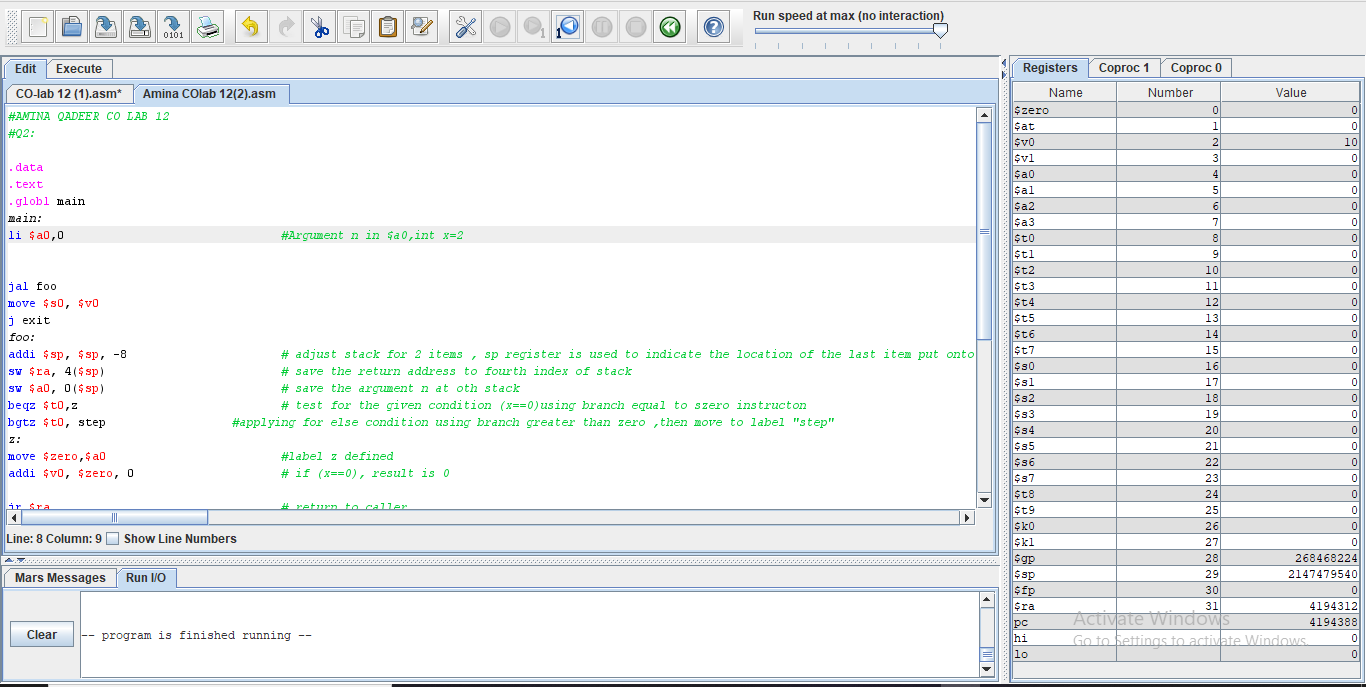
i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#the ijr iinstruction iis ifollowed iby ia ibranch idelay

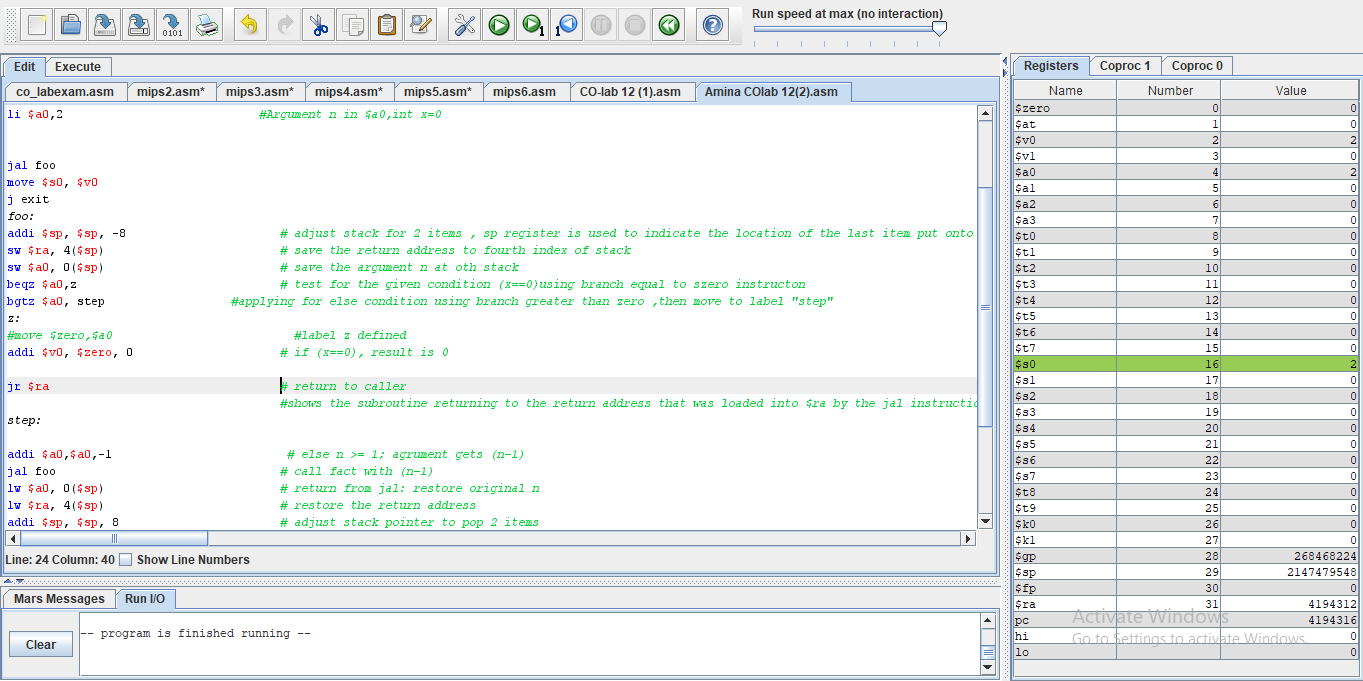
exit: i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#defining iExit ilabel ifor itermination iof icodedefining iExit ilabel ifor itermination iof icode

li i i$v0,10

syscall

**OUTPUT:**



****

**i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i ii i iiTHE i i i i i i i i i iiEND**