

Project III

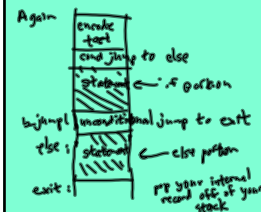
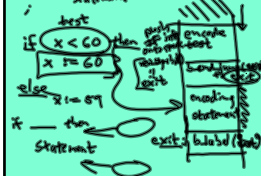
Control flow
Array accesses
Local vars

Control flow:
→ new-symbol()
→ b_label()
→ b_jump(label)
→ b_cond_jump(....)

if-statement:
simple if LEV ELSE statement

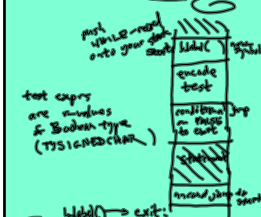
(simple if %prec --)

simple if:
LEV IF boolean-expression LEV THEN
statement



if - then
else if - then

while <test> do
statement



test expr
are numbers
& Boolean type
(TYPEDCHAR)

boolean-expression:
expression ()

FOR statements

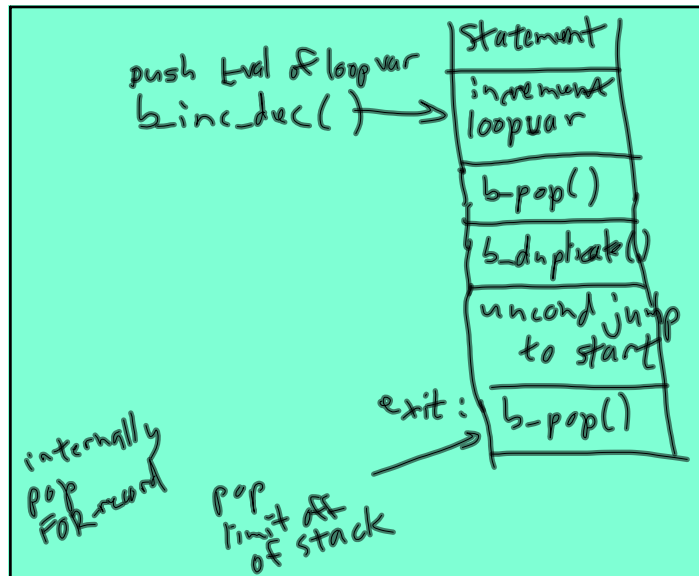
upper init limit
for i: init to limit do
statement

loop var, init, limit
are all of the same ordinal type
loop var is on level
init, limit are Real-val.

can convert all arguments to integer
initialization sequentially:
double and
precision
push and internal
formal parameter
onto your stack

Boolean
limit
limit

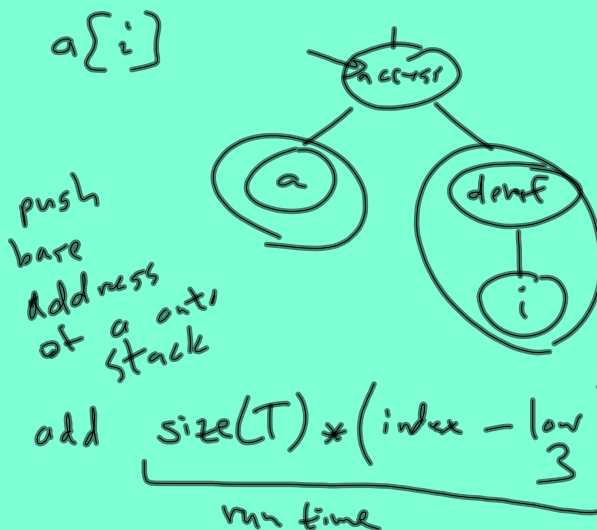
Boolean val
duplicate
limit
Boolean val of
test
fresh
duplicate
statement
test
returning value



b-dispatch() ← alternative to duplicating

for i := 6 downto 3 do

a : array [5..7] of (T)



new address
 (L-value of a[i])
 (i-value)

b-ptr-arith-op()