Heap Allocation

Eagle: dynamically allocate all non-primitive values

At program start: call malloc and allocate slab of memory

Global variable tracking next free byte of heap

C++

~~Foo foo; static allocation~~

Foo* foo; dynamic allocation

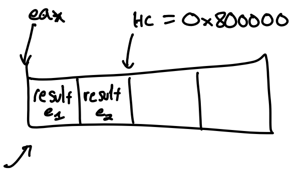
(4, 5)

```
mov eax, 8
mov [ebp-4], eax
mov eax, 10
mov [ebp-8], eax
mov ecx, [heap-cursor]
mov eax, [ebp-4]
mov [ecx], eax
mov eax, [ebp-8]
mov [ecx+4], eax
mov eax, ecx
add ecx, 8
mov [heap-cursor], ecx
or eax, 1
```

bird pointers end in 01

machine pointers end in 00

heap_cursor is a machine ptr



0x80000008 = 0x8000000[1000]

0x8000000[1001] = 0x80000009

let x = (4,5) in first(x)

eax

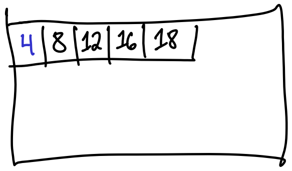
all the stuff from above

```
mov [ebp-4], eax
mov eax, [ebp-4]
and eax, 0xFFFFFFFC      xor eax, 1      dec eax
mov eax, [eax]
```

Eagle

(4, 6, 8, 9)[8]

```
<expr> ::= .....
         | (<expr>, ..., <expr>)
         | <expr>[<expr>]
```



expected a tuple   4[3] : err code 4

index oob .   (1,2)[-1] : err code 5