

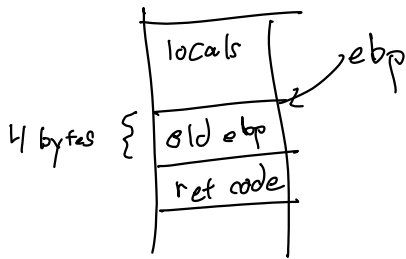
Pairs

- take more than 32-bits!
- store pairs in heap memory (dynamically allocated)
- change the binary repr to include pers
 - ints end in 0
 - bool end in 11
 - ptrs end in 01
- allocate heap w/ malloc - pass that to snake_main
- heap_cursor

```

C++
-----
Foo Foo;
Foo* foo;
  
```

0000 — 0000 1010 ^{repr} 5



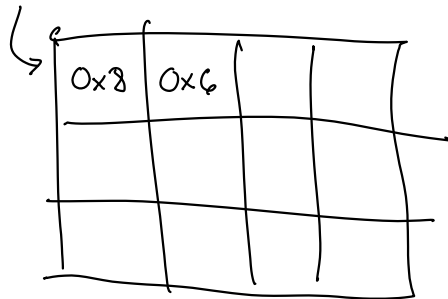
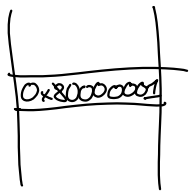
```

let x = 234567890 in
add-both-parts(x)
  
```

forbid this

point only to memory on 4-byte boundaries

0x80000000



Example: Creating a Pair

(4, 5)

let \$1 = 4 in
let \$2 = 5 in
(\$1, \$2)

EPair(EInt 4, EInt 5)

ALet("\$1", ACEXpr(CIEXpr(IInt 4)) ,
ALet("\$2", ACEXpr(CIEXpr(IInt 5)) ,
ACEXpr(CPair(IVar "\$1", IVar "\$2"))))

```
mov eax, 8  
mov [ebp-4], eax  
mov eax, 10  
mov [ebp-8], eax
```

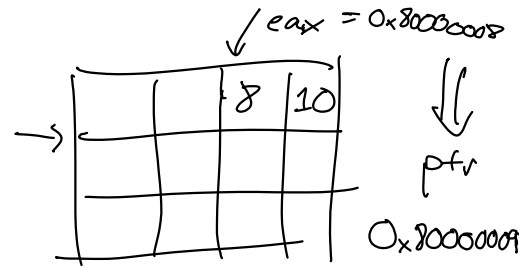
} let \$1 = 4 in

} let \$2 = 5 in

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

} (\$1, \$2)

or eax, 1



Pair Access

snd ((4, 5))

let \$1 =

let \$2 = 4 in

let \$3 = 5 in

(\$2, \$3)

in

snd (\$1)

add eax, 1

mov [ebp-12], eax

mov eax, [ebp-12]

sub eax, 1

mov eax, [eax+4]

snd

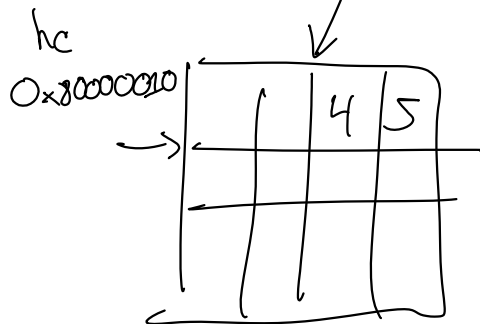
<expr> ::=

| (<expr>, <expr>)

| fst(<expr>)

| snd(<expr>)

0x80000008



eax = 0x80000009

Tuples

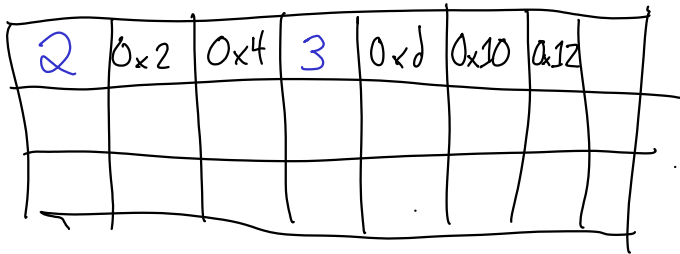
let $x = (1, 2)$ in

let $y = (7, 8, 9)$ in

let $z = \text{if } b \text{ then } x \text{ else } y$ in
 $z[2]$

Tuple Heap Layout

- 32-bits: size of tuple \leftarrow just a number
- N 32-bit snake values: elements



Tuple Problems? $x \geq 3$

$(1, 2, 3)[x] \Rightarrow$ runtime error 5

$(1, 2) * 0 \Rightarrow$ rt err 1

$(1, 2) \&\& \text{true} \Rightarrow$ rt err 2

$\text{true}[0] \Rightarrow$ runtime error 4

$(1, 2)[\text{true}] \Rightarrow$ rt. err 1

