/\* memory \*/

/\*

\* pointers

\*/

ptr \*void = nil // points nowhere

ptr \*void = --- // garbage value, points to random memory

ptr \*utf = 0x7FFFFFFF // points to 0x7FFFFFFF

// ‘&’ is the address\_of operator

\_ptr \*utf = &ptr // points to ptr

// ‘\*’ is the dereference operator

\*\_ptr = ‘x5F1D’ // here we set the value reference by \_ptr to a new value

/\*

\* getting memory from the OS

\* and returning memory to the OS

\*/

ptr \*u64 = os.get\_mem(47, u64) // get a memory block of size 47 \* 8(size of u64) bytes

os.ret\_mem(ptr) // return memory pointed by ‘ptr’ to the OS

/\*

\* simple stack array

\*/

array \*s32 = {} // empty array, UNSUPPORTED! use array \*s32 = nil

array \*s32 = { 14, 957, -5, 0, -4096 } // array of size 5 initialized

array \*s32 = [7]{num} // array of size 7 initialized at num

array \*s32 = [7]{---} // array of size 7 unitialized

/\*

\* access elements (works with any pointer)

\*/

array[0] // first element of the array

array[1] // second element of the array

.

.

.

array[n-1] // n-th element of the array