Assignment 5 onwards - Per function documentation on code
Tutorial 6 - psct 15 to 17
Lab 6 - Exercise on call by reference & dynamically allocated arrays

Variables

- Location/Place in computer's memory that holds one or more variables.
- we can access variable through its memory address.

Address-of operator (&)

If x is the variable, &x is its memory address.
If x has the type T, &x has the type $\underbrace{*}_{\text{pointer to T}}$T

cs1010_println_pointer   prints the pointer/memory address of a particular variable

Dereference operator (*)

If x is a memory address ~~*~~ *x is the content at memory address x.
If x has the type *T, *x has the type T.

```
int main(){
    double *addr;          //pointer-to-double
    double d = 100.0;
    addr = &d;             // assigns memory address of d to pointer var addr
    cs1010_println_double(*addr);  // print value stored at memory address
                                      addr.
}
```

address-of

dereference

[illegible]

Rules of pointers

- Pointers must be of the same type as the variable it references

  void * is a pointer to anything

- Pointer arithmetic
  - addition adds multiple of the size (in bytes) of the type it is pointing to
  - $a[i] = *(\boxed{a} + i)$

    &a[0]

- Pointers are variables too

  *T → T (*T is a pointer to T)

  T → *T (T is a pointer to the pointer T)

- NULL pointer, pointer that points to "nothing".

CALL BY REFERENCE

swap

```
void swap(long *a, long *b){
      long temp = *a;
      *a = *b;
      *b = temp;
}
```

follows pointer to a and dereferences it to obtain value stored in a. temp is initialized to hold this value

dereferences value at b and follows pointer to mem address containing a. Assigns the value b to this address

assign value b is pointing at to value stored at temp.

Heap

- Program must explicitly "borrow" memory space for use
- Program must "return" the memory space after use
- Lifetime of content stored on heap can exceed the lifetime of the function that allocates it.

Limit on heap >> limit on stack

How to allocate memory from heap?

malloc() - memory allocation function
- allocates size bytes and returns a pointer to the allocated memory

sizeof() - returns "size" of type

free() - function that deallocates memory in heap ("returns" memory after use).

calloc() can also be used.

Allocate memory for array based on input

```
size_t n = cs1010_read_size_t();
double *ar = malloc (n*sizeof(double));
.
.
.
free (ar);
```

CHARACTERS & STRINGS

NULL character == '\0'

A string is a character array that is terminated by the special NULL character. Allocate (n+1) space for n-char string.

String literals

stored in read-only region of memory

$\downarrow$

"text" region