1. Array of int/char
2. Array of 2D int/chars
3. Array of structs
   1. Example for a record struct
      1. record\* = malloc(n\*sizeof(record));
      2. record\* = calloc(n, sizeof(record));
         1. //record is a ptr to an array of size n-1
      3. struct(record[i].name, “John Doe”);
4. Array of ptr to structs
   1. Example of an array of ptr to struct
      1. record \*\* = malloc(n\*sizeof(record\*));
         1. //record is a ptr to ptrs
      2. for (i = 0; i < n; i++)
         1. record[i] = NULL;
         2. //Important to initialize to NULL
      3. for (i = 0; i < n; i++)
         1. record[i] = malloc(sizeof(record));
      4. strcpy(record[i]->name, “John Doe”);
         1. //-> means \*record[i].name
   2. To free
      1. for (i = 0; i < n; i++)
         1. free(record[i]);
      2. free(record);
      3. //Every malloc() has an equal and opposite free()
5. ptr to struct that has a ptr
   1. big int \* num = malloc(sizeof(big int))
      1. num 🡨 🡪 digits, size
      2. /I want to copy the contents of str (char\*) into digits
      3. num->size = strnlen(digits);
      4. num->digits calloc(numsize, sizeof(int));
      5. for (i = 0, i < num->size, i++)
         1. digits[i] = str[i] = 0;
   2. To free
      1. free(num->digits);
      2. free(num)