1. malloc()
   1. Name of array is a pointer to a memory address
   2. scanf(“%d”, &arr1[i]) 🡨🡪 scanf(“%d”, arr1 + i);
   3. How to allocate an array dynamically so everything is zeroed out
2. calloc()
   1. Zeroes out the arrays
   2. Format
      1. int \* arr2 = calloc(n, sizeof(int));
         1. //n = # of elements
         2. //sizeof(int) = size of each element
   3. Storage
      1. char typically written in 2 bytes
      2. int typically written in 4 bytes
      3. In memory, shifts are by 4 bytes.
      4. Storage values may be overwritten if not enough memory is allocated.
3. realloc()
   1. Used when there is not sufficient memory
   2. Format
      1. arr1= realloc(arr1, numbytesnew));
         1. //arr1 = old ptr 🡪 new ptr
         2. //numbytesnew = number of bytes needed
4. free()
   1. Used to free memory
5. Why DX/DY arrays are nice (Word Search Problem)
   1. Code
      1. for (i = 0; i < 8; i++) {
      2. //Check word in direction i
      3. //Look at indexes
         1. //(x+y), (x+DX[i], y+DY[i]), (x+2DX[i], y+2DY[i])…
      4. //To move char to char….
         1. //tempx += DX[i];
         2. //tempy += DY[i];
   2. Better than writing >3-point if-else-if-else statements