Assignment 2 releasing soon

End of this week or early next week

- Linked lists
- Dynamic memory
- Structs

Remember to get support

- Revision Sessions
- Help Sessions
- See forum for details

Memory Recap

`malloc()`

- `malloc` -> Memory Allocation (allocate memory on the heap)
- Returns a pointer to the location on the heap
- We can decide how large the allocation

Calling `malloc`

– `ptr = (cast-type*) malloc(byte-size)`

Example:

```c
#include <stdio.h>

int main(void) {
    malloc(1000);
    malloc(sizeof(int));
    malloc(sizeof(char) * 50);

    return 0;
}
```

Heap memory cheat sheet

- Allocate memory: `malloc()`
- Deallocate: `free()`
- Grow/shrink memory `realloc()`
- All require `stdlib.h`

  `sizeof()`

Dynamic arrays on the heap

A common way of using malloc is to create dynamic arrays:

```
int main(void) {
    int num_elements;
    scanf("%d", num_elements);

    int *data = malloc(num_elements *
sizeof(int));
    data[0] = 5;

    return 0;
}
```

Linked Lists

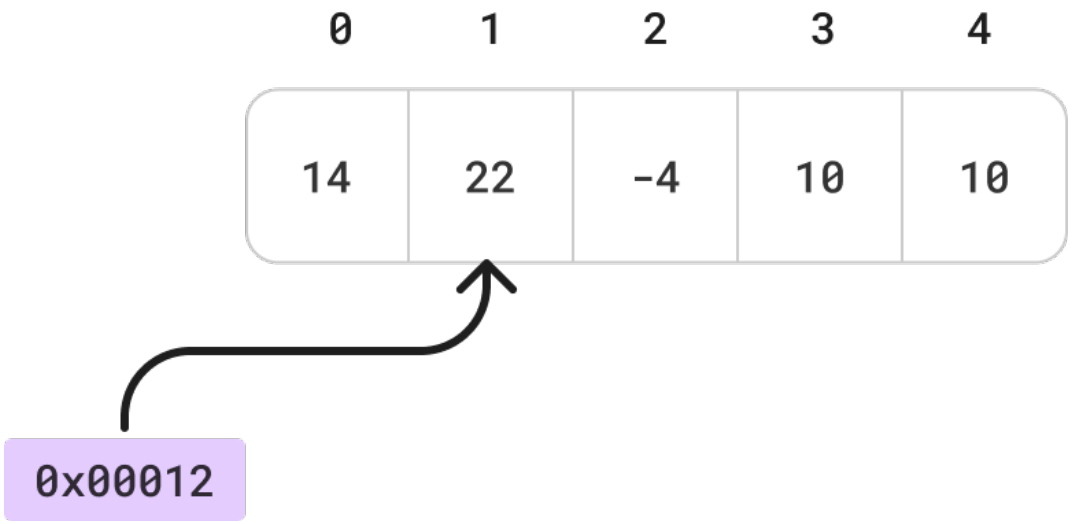
So far

arrays to store collections of data

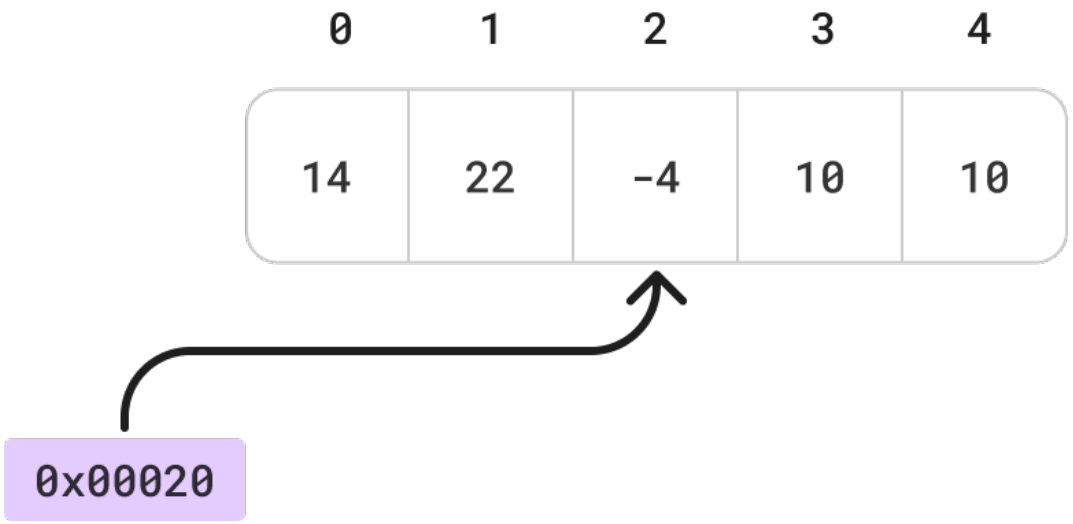
Arrays are contiguous, so we use the address of the first index to access each element

array variable points to start

Move along the `sizeof(int)`



Move along the `sizeof(int)`



Limitations of arrays

- If we know exactly how many elements we need to store, and we have the data, great!
- else, we need to have sufficient memory set aside in advance, or grow it, but...
- Allocating memory is **expensive**

Allocating memory is a bit light seat allocations, it's slow, messy and annoying. If we overbook a flight, what happens?

What if we had a way to store additional data very easily?

Where growing memory was cheap

Enter the linked list

Linked lists

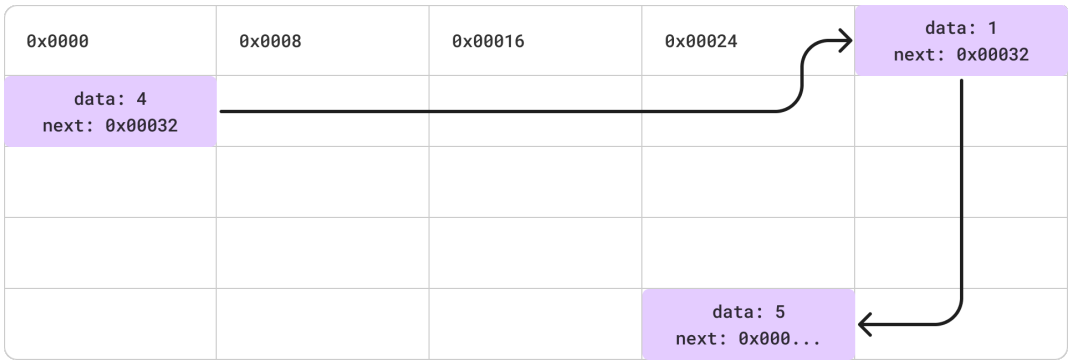
- Similar to dynamic arrays
  - they store collections of data
  - are dynamic (can grow/shrink)

Linked lists

- Different to arrays
  - *Efficiently* dynamic (you can add memory bit by bit)
  - are not contiguous
  - are not random access

Array

Linked List


0x0000
0x0008
0x00016
0x00024
data: 1
next: 0x00032
data: 4
next: 0x00032
data: 5
next: 0x000...

We use a struct on the heap

```
data: 4
next: 0x00032
```

```
struct node {
  int data;
  struct node *next;
};
```

Break, Kahoot, Demo

Demo goals

- Create a linked list with the elements `11, 8, 7`
- A reference to the linked list on the heap in `main`
- A way to print each element

Feedback

https://forms.office.com/r/K3PjvWebtD