Day 23 Highlights

1. Reminders
   1. zyBooks Chapter 8 due today at midnight
   2. Project 4 due this Friday at 5pm
   3. Exam 2 tomorrow, 5:00-6:20pm
   4. No lab this week
2. **malloc** (allocating space on the heap instead of stack) and **free**

**(type cast) malloc (size)**

1. Start with the following example

**#include <stdio.h>**

**#include <string.h>**

**void combineNames(char first[], char last[], char full[])**

**{**

**strcpy(full, first);**

**strcat(full, " ");**

**strcat(full, last);**

**// sprintf(full, "%s %s", first, last);**

**}**

**int main(void) {**

**char first[21];**

**char last[21];**

**printf("Enter first name: ");**

**scanf("%s", first);**

**printf("Enter last name: ");**

**scanf("%s", last);**

**char fullName[42];**

**combineNames(first, last, fullName);**

**printf("The full name is '%s' with a length of %d\n",**

**fullName, (int)strlen(fullName) );**

**return 0;**

**}**

1. Modify the function to return full name with return value
2. Arrays (one-dimensional arrays or 1D arrays)
   1. Can declare arrays either on the stack or on the heap

**int array1[100];**

**int \*array2;**

**array2 = malloc(sizeof(int) \* 100);**

* 1. When defining functions, can use [ ] or \* notation

**int funct1(int \*array, int size) {}**

**int funct1(int array[], int size) {}**

**in main: x = funct1(array, 100);**

* 1. Rewrite the functions of 1D arrays (using minmax.c of Day 22, March 1)
  2. Use **free** to free the memory allocated by malloc