COMP 222
C Programming Project
Ungraded Warm Up Exercise

1. Read in and store maintain values in a statically sized array.
2. Maintain values in a dynamically sized array.
3. Generate random numbers into the array and sort them.
4. Read numbers from a file and sort them.
5. Move operations into separate functions.

1. Read in and store values in a statically sized array.

In C, static arrays are sized when they are declared. This kind of array doesn't exist in Java.

```
int data[10];     // statically allocated array of 10 integers.
```

Once an array has been created we can read values in from the keyboard using scanf inside a loop:

```
int i;
…
for (i=0; i<10; i++) {
      printf("Enter value: ");
      scanf("%d",&data[i]);
}
```

Using the C compiler in "C99" mode allows you to use an updated version of C that accepts a few C++ and Java like syntax features.

2. Dynamically sized array

A dynamic array in C is more similar to an array in Java. You declare only a pointer but allocate the space only at runtime.

```
#include <stdlib.h>
…
int *data;
data = malloc(10*sizeof(int));
```

Once the dynamic array has been created, it can be used as if it had been declared as a static array.

```
data[3] = 5;
```

It can also be indexed using pointer arithmetic:

```
*(data + 3) = 5;
```

but static array indexing is usually easier to read. Unlike Java, C has no garbage collector so you have to free dynamically allocated space after use to avoid memory leaks.

```
free(data);
```

3. Generate random numbers into the array and sort them.

Use the srand() and rand() functions to generate random numbers. The srand() function is the seed function to initialize rand(). It's common practice to use the system clock via the time() function to create a different seed value each time the program is run. To reproduce the same list of numbers for different runs, pick a specific seed value and reuse it.

```
srand(time(0));
…
int i;
for (i=0; i<10; i++) {
        printf("%d ",rand()%1000);
}
```

Put the random values into an array, then print them out. Now use bubble sort or other algorithm to sort them and print them out again.

4. Read numbers from file

You can open a file and read from it in C by using a FILE pointer.

5. Move operations into separate functions.

You can define specialized functions in C just like you can define multiple class methods in Java. In C, all functions are global. Additionally, there is a **forward reference problem**, which in short means that you should declare a function before calling it as you read your source code from top to bottom.

This function takes two arguments: an int array and an int that gives the size of the array. This is standard practice in C when passing arrays as parameters, you'll usually need a second parameter to give the size of the array, since the size property is not built into the array like it is in Java.

```
void show(int d[], int sz) {
        …
}
```

Then call the function from another function like main and pass in the parameters.

```
int main(void) {
        int data[5] = { … };
        show(data,5);

        return 0;
}
```

Program Listings

1.

```
#include <stdio.h>

int main(void) {
      int data[10];

      for (int i=0; i<10; i++) {
            printf("Enter next value:  ");
            scanf("%d",&data[i]);
      }

      printf("Values are: ");
      for (int i=0; i<10; i++) {
            printf("%d ",data[i]);
      }
      printf("\n");
      return 0;
}
```

2.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {

      int *data;
      data = malloc(10*sizeof(int));
      int i;
      for (i=0; i<10; i++) {
            printf("Enter value: ");
            scanf("%d",&data[i]);
      }

      for (i=0; i<10; i++) {
            printf("%d ",data[i]);
      }
      printf("\n");

      free(data);

      return 0;
}
```

3.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
      srand(time(0));
      int *data;
      data = malloc(10*sizeof(int));
      int i;
      for (i=0; i<10; i++) {
            data[i] = rand()%1000;
      }

      for (i=0; i<10; i++) {
            printf("%d ",data[i]);
      }
      printf("\n");

      return 0;
}
```

4.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
  FILE *fp = fopen("numbers.txt","r");
  int i;
  int x;
  for (i=0; i<10; i++) {
    fscanf(fp,"%d",&x);
    printf("%d ",x);
  }
  printf("\n");

  return 0;
}
```

5.

```
#include <stdio.h>
#include <stdlib.h>

void show(int d[],int sz) {
      int i;
      for (i=0; i<sz; i++) printf("%d ",d[i]);
      printf("\n");
}

int main(void) {
      int data[] = {5,18,22,6,9};
      show(data,5);
      return 0;
}
```