Lab #5

CS-2050

September 30, 2022

1 Requirements

In this lab, you will cover working with structs and struct pointers. Additionally, you will be reviewing pointer arithmetic. You are provided the following struct definition to use with your starter code:

```
typedef struct {
        int lastCalculation;
        int size;
} ArrayInfo;
```

1.1 createArray

```
int * createArray(int size);
```

Info: This function will take an integer representing the size of the integer array to allocate. It will then allocate an integer array with the given size, and if successful, it returns a pointer to the array with an ArrayInfo struct hidden before the array pointer. Otherwise it will return NULL.

1.2 getSize

```
int getSize(int *array);
```

Info: This function will take an integer array with the array info hidden before the array pointer, and return the size of the array.

1.3 getInfo

```
ArrayInfo * getInfo(int *array);
```

Info: This function will take an array with the array info hidden before the array pointer as an ArrayInfo struct, and return a pointer to that struct.

1.4 printLastCalculation

```
void printLastCalculation(int *array);
```

Info: This function will take an integer array with the array info hidden before the array pointer as an ArrayInfo struct, and will print the value of the last calculation performed on the array, or -1 if one does not exist (you may assume that no calculation will result in -1).

Info: Example output for the print function:

```
// No previous calculation performed
Last calculation: -1
// Previous calculation resulted in 5
Last calculation: 5
```

1.5 sumFours

```
void sumFours(int *array);
```

Info: This function will take a pointer to an array with the array info hidden before the array pointer as an ArrayInfo struct, calculate the sum of all the fours in the array. The result of the calculation should be stored in the array info.

1.6 freeArray

```
void freeArray(int *array);
```

Info: This function will take a pointer to an integer array with the array info hidden before the array pointer as an ArrayInfo struct, and free the memory allocated to it.

Notice

Grading: 14 points

1. Write required *createArray* function
   * 4 points
2. Write required *getSize* function
   * 2 points
3. Write required *getInfo* function
   * 2 points
4. Write required *print* function
   * 2 points
5. Write required *sum* function
   * 2 points
6. Write required *free* function
   * 2 points