

Code Jam 2

Names:

Complete the following exercises in 40 minutes. This activity is open book, open computer. All work should be your own group.

Question 1

When Cedric runs his program with valgrind, he gets the following error.

```
==2552== Invalid read of size 4
==2552==    at 0x109276: main (memerr2.c:12)
==2552== Address 0x4a51040 is 0 bytes inside a block of size 40 free'd
==2552==    at 0x483CA3F: free (in /usr/lib/x86_64-linux-gnu/valgrind
/vgpreload_memcheck-amd64-linux.so)
==2552==    by 0x109258: main (memerr2.c:10)
==2552== Block was alloc'd at
==2552==    at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind
/vgpreload_memcheck-amd64-linux.so)
==2552==    by 0x1091EC: main (memerr2.c:6)
```

Here is his code. What does the valgrind error mean and how can Cedric fix his program? Write your answer in the space above, next to Cedric's program.

```
int main() {
    int n = 10;
    int *values = malloc(sizeof(int) * n);
    for (int i = 0; i < n; i++) {
        values[i] = rand() % 10 - 5;
    }
    free(values);
    for (int i = 0; i < n; i++) {
        printf("%d ", values[i]);
    }
    printf("\n");
    return 0;
}
```

Question 2

Consider the program, `linkedlist.c`, which is checked into your Jam directory.

```
struct cake {
    float cost;
    char flavor[16];
    struct cake* next;
};

struct cake makeCake(const char* flavor, float cost) {
    struct cake newCake;
    strncpy(newCake.flavor, flavor, 16);
    newCake.cost = cost;
    newCake.next = NULL;
    return newCake;
}

int main() {
    struct cake cake1 = makeCake("red velvet", 2.00);
    struct cake cake2 = makeCake("chocolate", 1.75);
    struct cake cake3 = makeCake("mocha", 3.50);

    cake1.next = &cake2;
    cake2.next = &cake3;
    // A) draw stack and heap here
}
```

A) In the space above, draw the function stack for the following program at the end of `main()`. You *do not* need to include intermediate values in your diagram — just include the state of the stack and heap. Please do not use CTutor.

B) Does the above program have memory leaks? Why or why not?

D) In the file, `linkedlist.c`, add a fourth cake the end of the list.

E) In the file, `linkedlist.c`, implement a function `cheapestCake` that prints each cake followed by the cheapest one. Below is an example of possible output. Your function should take the first cake as an argument.

```
$ make linkedlist
gcc -g linkedlist.c -o linkedlist
$ ./linkedlist
cake: red velvet ($2.00)
cake: chocolate ($1.75)
cake: mocha ($3.50)
cake: rainbow sprinkle ($5.25)
The cheapest cake is chocolate
```

Question 3

Write a program, `readvector.c`, that implements a function that reads in a vector from a file and prints the result.

A vector is a N-dimensional list of components. For example, an example of a 3-dimensional vector is (0.5, 6, -7.1).

Vectors are stored in a file, with one value per line. The first value in the file stores the length of the vector. Each subsequent line stores the corresponding values.

```
$ cat vector1.txt
3
0.5
6
-7.1
$ make readvector
gcc -g readvector.c -o readvector
$ ./readvector vector1.txt
0.500000
6.000000
-7.100000
```

Requirements:

- Your program should run without memory errors.

- You should use `fopen/fclose` to open and close the file.
- You should not change the function signatures in your base code.
- Your program should read the filename as a command line argument.
- Use `malloc/free` to allocate memory for your vector (do not assume a max size!)
- You can assume no number requires more than 32 characters to store.
- Hint: use `atof` to convert from a string to a float value

Last updated 2022-02-24 12:02:44 -0500