Name: Sean Losseff RCS ID: losses @rpi.edu

CSCI 4210 — Operating Systems ///
Spring 2020 Quiz 1 (January 26, 2020)

- Please silence and put away all laptops, notes, books, phones, electronic devices, etc.
- This quiz is designed to take 25 minutes; therefore, for 50% extra time accommodations, the expected time is 38 minutes and 100% extra time accommodations is 50 minutes (i.e., the end of class).
- Questions will not be answered except when there is a glaring mistake or ambiguity in a question. Please do your best to interpret and answer each question.

1. (4 POINTS) What is the exact terminal output on stdout for the code below? Ignore stderr output. Circle the best answer.

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

int main()
{
  float * f = NULL;
  printf( "ABCDEFGHIJKLM" );
  printf( "NOPQRSTUVWXYZ\n" );
  *f = 1.24;
  printf( "WOW\n" );
  return EXIT_SUCCESS;
}
```

(a) ABCDEFGHIJKLM\NOPQRSTUVWXYZ\nWOW\n
(b) (no output)
(c) ABCDEFGHIJKLMNOPQRSTUVWXYZ\n
(d) ABCDEFGHIJKLM

2. (3 POINTS) When you use `malloc(10 * size_of(int * ))`, how is memory allocated? Circle the best answer.

(a) 40 bytes are allocated on the runtime stack.
(b) 40 bytes are allocated on the runtime heap.
(c) 80 bytes are allocated on the runtime stack.
(d) 80 bytes are allocated on the runtime heap.

3. **(3 POINTS)** When you call `realloc(ptr, new_size)`, what is guaranteed? Circle the best answer.

   (a) Realloc always succeeds.
   (b) The new memory is initialized to 0.
   (c) The values stored in ptr are all copied into the new memory.
   (d) The memory pointer ptr remains valid.
   (e) None of (a)-(d) is guaranteed.

4. **(10 POINTS)** Write the **exact** terminal output of the given code in the space below. Assume a 64-bit architecture and that all system calls return successfully. Note that there are no compilation warnings or errors in the given code.

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

int main()
{
  char * name = "CSCI\0-4210";
  char * q = calloc( 10, sizeof( char ) );
  strcpy( q, name );
  q[5] = '0';
  q[6] = 'S';
  printf( "%d", (int) strlen( name ) );

  int j;
  fprintf( stderr, "Welcome to %s.\n", q );
  for ( j = 0 ; j < 4 ; j++ )
  {
    printf("E");
    printf( "\n");
  }
  return EXIT_SUCCESS;
}
```

q = | C | S | C | 1 | 10 | 0 | S | 2 | 1 | 0 |

strlen(name) = 5

welcome to CSC1. \n

Output

5 Welcome to CSC1. \n

E \n

E \n

E \n

E \n