```cpp
#include "main.h"
#include "init.h"
#include "portdef.h"

// Character array holdign the vaious labels for the autonomousroutines - labels are
// the function names being called
const char* titles[] = {"Skills", "autoRedLeft", "AutoBlueLeft", "autoRedRight", "autoBlueRight"};

int selection;                                                  // Select script to run
unsigned int scriptNumber = 0;                  // scriptNumber which will be passed

static bool selectionMade = false;          // Sart fresh - no selection made yet

// LEFT lcd display button call back routine
// Button pressed decrements in the selection array
void on_left_button() {
    static bool pressed = false;
  if(!selectionMade){
        pros::lcd::clear_line(4);
        if (scriptNumber != 0) {
            scriptNumber--;
          pros::lcd::print(2, "Script#: %d\n", scriptNumber);
      pros::lcd::print(3, titles[scriptNumber]);
      } else {
          pros::lcd::print(2, "Script#: %d Can't decrement\n", scriptNumber);
          pros::lcd::print(3, titles[scriptNumber]);
      }
    }
}

// CENTER make a selection or deselect the choosen option
void on_center_button() {
    static bool pressed = false;
    pressed = !pressed;
    if (pressed) {
        selection = scriptNumber;
        pros::lcd::set_text(4, "<< SELECTED ! >>");
        selectionMade = true;
    } else {
        pros::lcd::set_text(4, "<< DE-SELECTED >>");
        selection = 0;                  // Reset selection
    selectionMade = false;
    }
}

// RIGHT button, increments the array index to next selection
void on_right_button() {
    static bool pressed = false;
  if(!selectionMade){
        pros::lcd::clear_line(4);
        if (scriptNumber < (NUM_SCRIPTS - 1)) {
            scriptNumber++;
          pros::lcd::print(2, "Script#: %d\n", scriptNumber);
          pros::lcd::print(3, titles[scriptNumber]);
        } else {
            pros::lcd::print(2, "Script#: %d Can't increment\n", scriptNumber);
          pros::lcd::print(3, titles[scriptNumber]);
        }
    }
}

/**
 * Runs initialization code. This occurs as soon as the program is started.
 *
 * All other competition modes are blocked by initialize; it is recommended
 * to keep execution time for this mode under a few seconds.
 */
```

```
68 void initialize() {
69     // Motor Setup
70     // GEARSET_36 -- RED
71     // GEARSET_18 -- GREEN (default)
72     // GEARSET_6 -- BLUE
73 
74     pros::Motor front_right_motor(FRONT_RIGHT_MOTOR_PORT, pros::E_MOTOR_GEARSET_18, true, pros::E_M
75     pros::Motor front_left_motor(FRONT_LEFT_MOTOR_PORT, pros::E_MOTOR_GEARSET_18, false, pros::E_MO
76     pros::Motor back_right_motor(BACK_RIGHT_MOTOR_PORT, pros::E_MOTOR_GEARSET_18, true, pros::E_MOT
77     pros::Motor back_left_motor(BACK_LEFT_MOTOR_PORT, pros::E_MOTOR_GEARSET_18, false, pros::E_MOTO
78 
79     pros::Motor tray_motor(TRAY_MOTOR_PORT, pros::E_MOTOR_GEARSET_18, false, pros::E_MOTOR_ENCODER_
80 
81     pros::Motor lift_motor(LIFT_MOTOR, pros::E_MOTOR_GEARSET_36, false, pros::E_MOTOR_ENCODER_DEGRE
82 
83     pros::Motor right_roller_motor(RIGHT_ROLLER_MOTOR, pros::E_MOTOR_GEARSET_06, true, pros::E_MOTO
84     pros::Motor left_roller_motor(LEFT_ROLLER_MOTOR, pros::E_MOTOR_GEARSET_06, false, pros::E_MOTOR_
85 
86     pros::lcd::initialize();
87     pros::lcd::set_text(1, "Shrek is coming");
88 
89     pros::lcd::register_btn0_cb(on_left_button);
90     pros::lcd::register_btn1_cb(on_center_button);
91     pros::lcd::register_btn2_cb(on_right_button);
92 }
93 
94 /**
95  * Runs while the robot is in the disabled state of Field Management System or
96  * the VEX Competition Switch, following either autonomous or opcontrol. When
97  * the robot is enabled, this task will exit.
98  */
99 void disabled() {}
100 
101 /**
102  * Runs after initialize(), and before autonomous when connected to the Field
103  * Management System or the VEX Competition Switch. This is intended for
104  * competition-specific initialization routines, such as an autonomous selector
105  * on the LCD.
106  *
107  * This task will exit when the robot is enabled and autonomous or opcontrol
108  * starts.
109  */
110 void competition_initialize() {}
111 
```