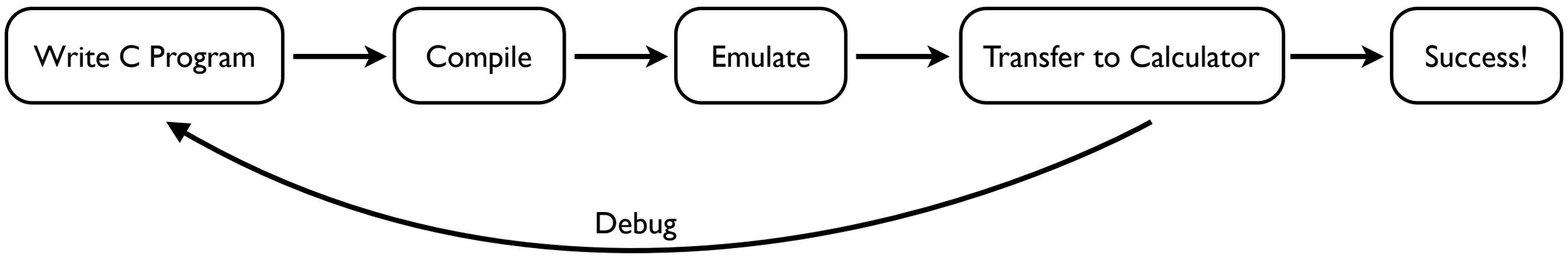
1. Install compiler
2. Install emulator
3. Install TI-Connect



Hello.c

```
#define SAVE_SCREEN
#define USE_TI89

#include <tigcclib.h>

_main() {
  clrscr();
  printf("Hello world!");
  ngetchx();
}
```

xray.c Part 1

```
BitmapPut(rect.xy.x0, rect.xy.y0, mask, &SCREEN, attr)
```

xray.c Part 2

```
switch (num) {
case 0:
  attr = A_AND;
  break;
case 1:
  attr = A_OR;
  break;
}
```

physics.c Part 1

```
while (1) {
  while (!kbhit()) {
    prevr = r;
    updateFunc(&r, &v);
    updateCircle(&r, A_NORMAL);
    updateCircle(&prevr, A_REVERSE);
  }
  k = ngetchx();
  switch (k) {
  // 2. *** YOUR CODE HERE ***
  default:
    return;
  }
}
```

physics.c Part 2

```
case KEY_F1:
  updateFunc = (void (*)(Vector*, Vector*)) updateLinearGravity;
  initialize(&r, &v);
  break;
case KEY_F2:
  updateFunc = (void (*)(Vector*, Vector*)) updateRadialGravity;
  initialize(&r, &v);
  break;
case KEY_CLEAR:
  initialize(&r, &v);
  break;
```

physics.c Part 3

```
case KEY_UP:
  v.y -= 1;
  break;
case KEY_DOWN:
  v.y += 1;
  break;
case KEY_LEFT:
  v.x -= 1;
  break;
case KEY_RIGHT:
  v.x += 1;
  break;
```

Extensions

- Easy: Two gravitational sources
- Easy: Have arrow keys change linear gravity
- Medium: Multiple bodies orbiting each other
- Hard: Better numerical methods such as Runge-Kutta
- Very Hard: Proper collision detection