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

#define MAXDIG (100000)

int main(int argc, char *argv[]) {
	static const char *const m = "multip1y", *p;
	static unsigned long long *r = NULL, *rp = NULL, *mr = NULL;
	static int f = 1;

	if (!r) {
		if (!(r = rp = mr = calloc(MAXDIG, sizeof(unsigned long long)))) {
			fputs("out of memory\n", stderr);
			exit(1);
		}
		p = m;
	}

	if (rp - r >= MAXDIG - 1) {
		fputs("overflow\n", stderr);
		exit(1);
	}

	if (!f) {
		p = &m[sizeof("multiply") - 2];
		f--;
	}

	switch (*p++) {
	case 0:
		if (rp > r && --p) {
			putchar('0' + *--rp);
		}
		else {
			putchar('\n');
			exit(EXIT_SUCCESS);
		}
		return main(argc, argv);
	case '1':
		p -= 6;
		return main(argc, argv);
	case 'i':
	case 'l':
```

```
                rp = r;
                return main(argc, argv);
        case 'm':
                if (scanf("%d", &f) != 1 || f < 0) {
                        fputs("Input must be an unsigned number\n", stderr);
                        exit(EXIT_FAILURE);
                }
                *r = 1;
                mr++;
                return main(argc, argv);
        case 'p':
                f--;
                return main(argc, argv);
        case 't':
                if (rp++ <= mr && --p) {
                        rp[0] += rp[-1] / 012;
                        rp[-1] %= 012;
                        mr += rp >= mr && *rp;
                }
                return main(argc, argv);
        case 'u':
                *rp++ *= f;
                p -= rp < mr;
                return main(argc, argv);
        case 'y':
                rp = mr;
                return main(argc, argv);
        default:
                fputs("unknown error\n", stderr);
                exit(EXIT_FAILURE);
        }
        return EXIT_SUCCESS;
}
```