Flash Cards

When I was younger my father used a program that produced random problems for me to solve for me to practice math. That was the first time I was motivated to dissect a program in listing form. Soon enough I was getting a perfect score regardless of how I did. To this day my programming skill outpaces my arithmetic.

This program is similar to that one. It will generate arithmetic problems for you to practice with. It also keeps a running total of how well you do and at the end of 25 problems will give you feedback both in percentage form and traditional letter grade.

The coolest feature of Flashcards is the way that it displays numbers and other symbols. Instead of simply putting up the problem it uses a sort of binary encoding to draw large numbers that fill the whole screen.

Certain screen can't see the problem when it's drawn up to down. If you run into that problem find the line:

```
v = (rand () < RAND_MAX / 2); strikes = 0;
```

and change it to:

```
v = 1; strikes = 0;
```

so that all the problems are always displayed horizontally.

Flash Cards is written by Joseph Larson

FLASHCARDS.C	You will need: a C/C++ complier .

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#define NUM 25

int shape[21][9] = {
  { 28,  54,  99,  99,  99,  99,  99,  54,  28},
  { 24,  28,  30,  24,  24,  24,  24,  24, 126},
  { 62,  99,  99,  48,  24,  12,   6,   7, 127},
  {127,  96, 112,  48,  56,  96,  99,  99,  62},
  { 32,  48,  52,  54,  54,  51, 127,  48,  48},
  {127,   3,   3,   3,  63,  96,  99,  51,  30},
  { 56,  12,   6,   3,  63,  99,  99,  99,  62},
  {127,  99,  48,  48,  24,  24,  12,  12,  12},
  { 28,  54,  54,  28,  54,  99,  99,  54,  28},
  { 62,  99,  99,  99, 126,  96,  48,  24,  14},
  {  0,  24,  24, 126, 126,  24,  24,   0,   0},
  {  0,   0,   0, 126, 126,   0,   0,   0,   0},
  {  0,   0, 102,  60,  24,  60, 102,   0,   0},
  {  0,  24,   0, 126, 126,   0,  24,   0,   0},
  {  0, 126, 126,   0,   0, 126, 126,   0,   0},
  {  0,   0,   0, 255, 255,   0,   0,   0,   0},
  {  0, 255, 231, 189, 153, 189, 231, 255,   0},
  {112, 140, 146,  80, 248, 136, 136, 132,   4},
  {  0,   0,  12,  96,  84, 234, 246,  73,  56},
  {128, 166, 146,  61, 139,  58,   6,   0,   0},
  {  0,   0,   0,   0,   0,   0,   0,   0,   0}};
char *name[21] = {"Zero", "One", "Two", "Three", "Four", "Five", "Six",
 "Seven", "Eight", "Nine", "Plus", "Minus", "Times", "Into", "Equals", "Equals",
 "Strike", "Right", "htRig", "ightR", " "};

void drawline (const int msg[6]) {
  int ch, col, row;

  for (row = 0; row < 9; row ++) {
    for (ch = 0; ch < 6; ch ++) for (col = 0; col < 8; col ++)
      putchar ((shape[msg[ch]][row] & 1 << col) ?
      name[msg[ch]][(col + row) % strlen (name[msg[ch]])] : ' ');
    putchar ('\n');
```

Listing continued on next page...


```
  }
  putchar ('\n');
}

int main (void) {
  int op, max, q1, q2, ans, input, strikes, s, v, z, line[6], score[4];
  double cscore;

  srand (time (NULL)); s = 0;
  printf ("Choose an option:\n 1) Addition (easy)\n 2) Addition (hard)\n"
    " 3) Subtraction (easy)\n 4) Subtraction (hard)\n"
    " 5) Multiplication\n 6) Division\n 7) Shuffle\n ? ");
  scanf("%d", &input);
  while (input < 1 || input > 7) {
    printf ("Please type a number between 1 and 7 ? "); scanf("%d", &input);
  }
  switch (input) {
    case 1 : max = 10; op = 1; break;
    case 2 : max = 99; op = 1; break;
    case 3 : max = 10; op = 2; break;
    case 4 : max = 99; op = 2; break;
    case 5 : max = 10; op = 3; break;
    case 6 : max = 10; op = 4; break;
    case 7 : max = 10; s = 1;
  }
  for (z = 0; z < 4; z++) score[z] = 0;
  for (z = 0; z < NUM; z++) {
    q1 = rand () % max; q2 = rand () % (max - 1) + 1; ans = rand () % max;
    v = (rand () < RAND_MAX / 2); strikes = 0;
    if (s) op = rand () % 4 + 1;
    switch (op) {
      case 1 : ans = q1 + q2; break;
      case 2 : if (q1 < q2) {q1 ^= q2; q2 ^= q1; q1 ^= q2;}
               ans = q1 - q2; break;
      case 3 : ans = q1 * q2; break;
      case 4 : q1 = ans * q2; ans = q1 / q2; break;
    }
    do {
      if (v) {
        line[0] = (q1 / 10) ? (q1 / 10) : 20;
        line[1] = q1 % 10;
        line[2] = 9 + op;
        line[3] = (q2 / 10) ? (q2 / 10) : 20;
        line[4] = q2 % 10;
        line[5] = 14;
        drawline (line);
      } else {
        line[0] = line[1] = line[4] = line[5] = 20;
        line[2] = (q1 / 10) ? (q1 / 10) : 20;
        line[3] = q1 % 10;
        drawline (line);
        line[0] = line[4] = line[5] = 20;
        line[1] = 9 + op;
        line[2] = (q2 / 10) ? (q2 / 10) : 20;
        line[3] = q2 % 10;
        drawline (line);
        line[0] = line[4] = line[5] = 20;
        line[1] = line[2] = line[3] = 15;
        drawline (line);
      }
```

Listing continued on next page...

FLASHCARDS.C	Listing continued from previous page...

```
      scanf("%d", &input);
      if (input != ans) {
        line[0] = line[2] = line[4] = 20;
        line[1] = (++strikes > 1) ? 16 : 20;
        line[3] = 16;
        line[5] = (strikes > 2) ? 16 : 20;
        drawline (line);
      }
    } while (input != ans && strikes < 3);
    line[0] = line[1] = line[4] = line[5] = 20;
    line[2] = (ans / 10) ? (ans / 10) : 20;;
    line[3] = ans % 10;
    drawline (line);
    if (input == ans) {
      line[0] = line[4] = line[5] = 20;
      line[1] = 17;
      line[2] = 18;
      line[3] = 19;
      drawline (line);
    }
    score[strikes] ++;
    puts ("Press any key...");
    getchar (); getchar ();
  }
  printf ("Report Card\n------ ----\n\n"
    "Out of %d:\n %d on first try (%2.1f%%)\n %d on second try (%2.1f%%)\n %d"
    " on last try (%2.1f%%)\n %d missed (%2.1f%%)\n\n",
    NUM, score[0], (float)score[0] * 100 / NUM,
    score[1], (float)score[1] * 100 / NUM,
    score[2], (float)score[2] * 100 / NUM,
    score[3], (float)score[3] * 100 / NUM);
  cscore = (float)score[0] * 100.0 / NUM + (float)score[1] * 50.0 / NUM +
    (float)score[2] * 25.0 / NUM;
  printf ("Grade: %c, (%2.2f%%)\n", 'A' + (cscore < 90.0) + (cscore < 80.0)
    + (cscore < 70.0) + 2 * (cscore < 60.0), cscore);
  exit (0);
}
```