**Приложение 3**

#include <Keypad.h>

#include <LiquidCrystal.h>

#include <math.h>

const byte ROWS = 5;

const byte COLS = 4;

const int len\_of\_exp = 24;

const int rs = 11, en = 10, d4 = 5, d5 = 4, d6 = 3, d7 = 2, N = len\_of\_exp \* 3, light = 13;

char str[N], kpol[N], moper[N];

int main () {

init();

bool flag\_of\_c\_e = 0, regime\_of\_c = 0, flag\_of\_full\_lcd = 0;

char key, x = 0, y = 0;

double res;

char c, i = 0, mi = -1, ki = 0, j = 0;

char keys[ROWS][COLS] = {

{'(', ')', '<', '.'},

{'1', '2', '3', '+'},

{'4', '5', '6', '-'},

{'7', '8', '9', '\*'},

{'C', '0', '=', '/'}

};

byte rowPins[ROWS] = {A1, A2, A3, A4, A5};

byte colPins[COLS] = {6, 7, 8, 9};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

lcd.begin(16, 2);

lcd.setCursor(x, y);

//Serial.begin(9600);

pinMode(light, OUTPUT);

digitalWrite(light, HIGH);

for (i = 0; i < N; i++) {

kpol[i] = 0;

moper[i] = 0;

str[i] = 0;

}

i = 0;

key = 0;

while (1) {

//Serial.println("new global cycle");

while (!flag\_of\_c\_e) {

key = keypad.getKey();

if (key != NO\_KEY) {

digitalWrite(light, LOW);

delay(300);

digitalWrite(light, HIGH);

if (key == 'C' || key == '=') {

flag\_of\_c\_e = 1;

} else {

if (!regime\_of\_c) {

if (key == '<') {

//delite symbol

if (i > 0) {

//Serial.println("<");

 i--;

str[i] = 0;

x = x - 1;

lcd.setCursor(x, y);

lcd.print(" ");

lcd.setCursor(x, y);

}

} else {

//add symbol

if (i < len\_of\_exp) {

if ((key == ')' && (str[i - 1] == ')' || (str[i - 1] >= 48 && str[i - 1] <= 57)))

|| (key == '(' && (i == 0 || str[i - 1] == 42 || str[i - 1] == 43 || str[i - 1] == 45 || str[i - 1] == 47))

|| (key == '.' && (str[i - 1] >= 48 && str[i - 1] <= 57))

|| ((key == '+' || key == '\*' || key == '/') && (str[i - 1] == ')' || (str[i - 1] >= 48 && str[i - 1] <= 57)))

|| (key == '-' && (i == 0 || (str[i - 1] >= 48 && str[i - 1] <= 57) || str[i - 1] == ')' || str[i - 1] == '('))

|| ((key >= 48 && key <= 57) && (i == 0 || str[i - 1] == '(' || str[i - 1] == '+' || str[i - 1] == '\*' || (str[i - 1] >= 45 && str[i - 1] <= 57)))

) {

if (i == 15 && !flag\_of\_full\_lcd) {

flag\_of\_full\_lcd = 1;

lcd.clear();

for (c = 8; c < 16; c++)

lcd.print(\*(str+c));

x = 7;

lcd.setCursor(x, y);

}

//Serial.println(key);

str[i] = key;

i++;

lcd.print(key);

x = x + 1;

}

}

}

}

}

}

}

if (key == 'C') {

//Serial.println("C");

lcd.clear();

for (i = 0; i < N; i++) {

kpol[i] = 0;

moper[i] = 0;

str[i] = 0;

}

x = 0; y = 0; i = 0; mi = -1; ki = 0; j = 0;

flag\_of\_c\_e = 0;

regime\_of\_c = 0;

key = 0;

}

if (key == '=') {

if (!regime\_of\_c) {

i = 0;

/\*Serial.println("=");

for (c = 0; c < N; c++) {

Serial.print(str[c], DEC);

Serial.print(' ');

}

Serial.println(' ');\*/

j = 0;

while (j != 4 && j != 5) {

j = return\_case(str, kpol, moper, i, ki, mi);

switch (j) {

 case 0:

//Serial.print("case 0\n");

case\_0(str, kpol, moper, i, ki, mi);

break;

case 1:

//Serial.print("case 1\n");

case\_1(str, kpol, moper, i, ki, mi);

break;

case 2:

//Serial.print("case 2\n");

case\_2(str, kpol, moper, i, ki, mi);

break;

case 3:

//Serial.print("case 3\n");

case\_3(str, kpol, moper, i, ki, mi);

break;

default:

break;

}

}

/\*Serial.println("kpol at main");

for (c = 0; c < N; c++) {

Serial.print(kpol[c], DEC);

Serial.print(' ');

}

Serial.println(' ');\*/

res = calc(kpol);

x = 0;

y = 1;

lcd.setCursor(x, y);

lcd.print("=");

lcd.print(res);

flag\_of\_c\_e = 0;

regime\_of\_c = 1;

key = 0;

}

flag\_of\_c\_e = 0;

}

}

}

char return\_case (char str[N], char kpol[N], char moper[N], char &i, char &ki, char &mi) {

/\*Serial.print("func return case...\nstr[");

Serial.print(i, DEC);

Serial.print("] = ");

Serial.println(str[i], DEC);\*/

if (str[i] >= 48 && str[i] <= 57 || str[i] == 46) {

return 0;

}

if ((moper[mi] == 0 && (str[i] == 40 || str[i] == 42 || str[i] == 43 || str[i] == 45 || str[i] == 47))

|| (moper[mi] == 43 && (str[i] == 42 || str[i] == 47 || str[i] == 40))

|| (moper[mi] == 45 && (str[i] == 42 || str[i] == 47 || str[i] == 40))

|| (moper[mi] == 42 && str[i] == 40)

|| (moper[mi] == 47 && str[i] == 40)

|| (moper[mi] == 40 && (str[i] == 40 || str[i] == 42 || str[i] == 43 || str[i] == 45 || str[i] == 47))

) {

return 1;

}

if ((moper[mi] == 43 && (str[i] == 10 || str[i] == 0 || str[i] == 43 || str[i] == 45 || str[i] == 41))

|| (moper[mi] == 45 && (str[i] == 10 || str[i] == 0 || str[i] == 43 || str[i] == 45 || str[i] == 41))

|| (moper[mi] == 42 && (str[i] == 10 || str[i] == 0 || str[i] == 41 || str[i] == 42 || str[i] == 43 || str[i] == 45 || str[i] == 47))

|| (moper[mi] == 47 && (str[i] == 10 || str[i] == 0 || str[i] == 41 || str[i] == 42 || str[i] == 43 || str[i] == 45 || str[i] == 47))

 ) {

return 2;

}

if (moper[mi] == 40 && str[i] == 41) return 3;

if (moper[mi] == 0 && (str[i] == 0 || str[i] == 10)) return 4;

if ((moper[mi] == 40 && (str[i] == 0 || str[i] == 10)) || (moper[mi] == 0 && str[i] == 41)) return 5;

}

void case\_0 (char str[N], char kpol[N], char moper[N], char &i, char &ki, char &mi) {

char c;

if (i > 0 && str[i - 1] == 45) {

kpol[ki] = 39;

ki++;

}

kpol[ki] = str[i];

ki++; i++;

/\*Serial.println("kpol at case\_0");

for (c = 0; c < N; c++) {

Serial.print(kpol[c], DEC);

Serial.print(' ');

}\*/

}

void case\_1 (char str[N], char kpol[N], char moper[N], char &i, char &ki, char &mi) {

kpol[ki] = 44;

ki++;

mi++;

moper[mi] = str[i];

i++;

}

void case\_2 (char str[N], char kpol[N], char moper[N], char &i, char &ki, char &mi) {

kpol[ki] = moper[mi];

ki++;

moper[mi] = 0;

mi--;

}

void case\_3 (char str[N], char kpol[N], char moper[N], char &i, char &ki, char &mi) {

i++;

moper[mi] = 0;

mi--;

}

double calc (char kpol[N]) {

//Serial.println("calc...");

double res[N], sum, drob;

char i = 0, i\_in\_res = 0, dec = 0, j;

//Serial.println("kpol at calc");

for (i = 0; i < N; i++) {

res[i] = 0;

//Serial.print(kpol[i], DEC);

//Serial.print(' ');

}

//Serial.println(' ');

i = 0;

bool flag\_of\_fault\_minus = 0, flag\_of\_minus\_num = 0, flag\_of\_num = 0;

while (kpol[i] != 0) {

//Serial.println("at while");

if (kpol[i] == 39) {

flag\_of\_minus\_num = 1;

i++;

}

if (kpol[i - 1] == 39 && i > 2 && kpol[i - 2] == 44 && kpol[i - 3] == 44) {

flag\_of\_fault\_minus = 1;

}

if (kpol[i] >= 48 && kpol[i] <= 57) { // compose the number

flag\_of\_num = 1;

res[i\_in\_res] = res[i\_in\_res] \* 10 + (kpol[i] - 48);

dec++;

 i++;

if (kpol[i] == 46)

dec = 0;

}

if (kpol[i] == 46) {

i++;

while (kpol[i] >= 48 && kpol[i] <= 57) {

drob = kpol[i] - 48;

res[i\_in\_res] += drob / pow(10, dec + 1);

dec++;

i++;

}

}

if (kpol[i] == 44) {

if (flag\_of\_num) {

if (flag\_of\_minus\_num == 1) {

res[i\_in\_res] \*= (-1);

flag\_of\_minus\_num = 0;

}

i\_in\_res++;

dec = 0;

flag\_of\_num = 0;

}

i++;

}

if ((kpol[i] < 48 || kpol[i] > 57) && kpol[i] != 44 && kpol[i] != 46 && flag\_of\_num == 1) {

if (flag\_of\_minus\_num == 1) {

res[i\_in\_res] \*= (-1);

flag\_of\_minus\_num = 0;

}

i\_in\_res++;

dec = 0;

flag\_of\_num = 0;

}

if (flag\_of\_fault\_minus == 1 && kpol[i] == 45) {

i++;

flag\_of\_fault\_minus = 0;

}

if (kpol[i] == 42) {

res[i\_in\_res - 2] = res[i\_in\_res - 2] \* res[i\_in\_res - 1];

res[i\_in\_res - 1] = 0;

i\_in\_res -= 1;

i++;

}

if (kpol[i] == 47) {

res[i\_in\_res - 2] = res[i\_in\_res - 2] / res[i\_in\_res - 1];

res[i\_in\_res - 1] = 0;

i\_in\_res -= 1;

i++;

}

if (kpol[i] == 43 || kpol[i] == 45) {

if (i\_in\_res >= 2) {

res[i\_in\_res - 2] = res[i\_in\_res - 2] + res[i\_in\_res - 1];

res[i\_in\_res - 1] = 0;

i\_in\_res -= 1;

}

i++;

}

for (j = 0; j < 10; j++) {

Serial.print(kpol[j], DEC);

Serial.print(" ");

}\*/

}

//Serial.println("aswer is: ");

//Serial.print(res[0], DEC);

return res[0];

}

