

' ДОБАВИМ ВОЗМОЖНОСТЬ ЧИСЛЕННОГО ДИФФЕРЕНЦИРОВАНИЯ

DECLARE SUB REC8 (R AS SINGLE, PX!, PY AS INTEGER)

DECLARE SUB REC (R AS SINGLE, PX!, PY AS INTEGER)

DECLARE FUNCTION CalcXp! (X1, Y1, X2, Y2, Yc)

5 SCREEN 12

10 COLOR 15

' ***** БЛОК ФУНКЦИЙ

M# = 10

B# = 5

pi# = 3.14159265358#

Tshift = 0

15 DEF fnu (X#)

' -----

fnu = SQR(ABS(X#+3))-COS(0.387*X#)

' -----

18 END DEF

' *****

'определение графического окна

VIEW SCREEN (0, 0)-(639, 273), 0, 2

20 VIEW PRINT 20 TO 24: CLS 2: INPUT " Xmin="; Xmin

21 INPUT " Xmax="; Xmax

22 CLS 1 '0

23 VIEW PRINT

24 '\$DYNAMIC

25 REDIM Y(530)

26 REDIM F(530)

27 REDIM DF(530)

'расчет коэффициентов отображения по X

40 Kx = 530 / (Xmax - Xmin): Bx = 630 - Kx * Xmax

'ЗАПОЛНЕНИЕ МАССИВА

45 FOR i = 100 TO 630 STEP 1

50 X = (i - Bx) / Kx

55 F(i - 100) = fnu(X)

60 NEXT i

GOSUB ChangeYF

'ВЫБОР Ymin И Ymax для автоматического масштабирования по Y

GOSUB ChoiceYM

'расчет коэффициентов отображения по Y

GOSUB CalcKBy

'определение шага по X

dx = 1 / Kx

'РАЗМЕТКА

GOSUB Grid

'график

COLOR 9

GOSUB Curve

146 **COLOR** 11

150 **LOCATE** 18, 15: **PRINT** "'V'-Метод Вилки 'R'-RESTART

'E'-Экстремум ' '-EXIT 'D'-дифференцирование"

155 **COLOR** 15

160 C\$ = **INKEY\$**

175 **IF** C\$ = "R" **OR** C\$ = "r" **THEN GOTO** 20

180 **IF** C\$ = " " **THEN GOTO** 210

185 **IF** C\$ = "V" **OR** C\$ = "v" **THEN GOSUB** VILKA

190 **IF** C\$ = "E" **OR** C\$ = "e" **THEN GOSUB** EXTR

195 **IF** C\$ = "D" **OR** C\$ = "d" **THEN GOSUB** Diff

196 **IF** C\$ = "M" **OR** C\$ = "m" **THEN GOSUB** MappY

200 **GOTO** 160

210 **SYSTEM**

'***** Конечная программа

'+++++++ Блок подпрограмм

+++++++

VILKA:

220 **COLOR** 13

225 **VIEW PRINT** 20 **TO** 24: **CLS** 2: **PRINT** " START VILKA"

231 **INPUT** " Xmin="; Xmn#: **INPUT** " Xmax="; Xmx#

IF fnu(Xmx#) * fnu(Xmn#) > 0 **THEN**

PRINT " Ошибка, на этом участке корней или нет, или их
несколько."

PRINT "Повторите, если желаете."

GOTO 280

END IF

240 **INPUT** " e="; E#

250 X0# = (Xmx# + Xmn#) / 2: W# = **ABS**(X0# - Xmx#)

255 **IF** fnu(X0#) * fnu(Xmx#) ≤ 0 **THEN** Xmn# = X0#

```

260 IF fnu(X0#) * fnu(Xmn#) ≤ 0 THEN Xmx# = X0#
265 IF W# > E# GOTO 250
270 PRINT " X0="; X0#
280 VIEW PRINT: COLOR 15
300 RETURN

```

EXTR:

```

305 COLOR 13
310 VIEW PRINT 20 TO 24: CLS 2: PRINT " START LMIN"
315 INPUT " Xmin="; Xmn#: INPUT " Xmax="; Xmx#: INPUT " Xsec="; Xsc#
320
330
395 COLOR 15
400 RETURN

```

Diff:

' Расчет кривой производной. Заполнение массива DF.

DF(0) = (F(1) - F(0)) / dx

DF(530) = (F(530) - F(529)) / dx

FOR i = 1 TO 529

DF(i) = (F(i + 1) - F(i - 1)) / (2 * dx)

NEXT i

COLOR 13

VIEW PRINT 20 TO 24: CLS 2

PRINT " Отобразить в одном поле оба графика Y/N? (или только производную) "

S\$ = ""

DO

S\$ = INKEY\$

LOOP UNTIL (S\$ = "y") OR (S\$ = "Y") OR (S\$ = "N") OR (S\$ = "n")

CLS 1 'очистка графического окна

IF (S\$ = "Y") OR (S\$ = "y") THEN ' рисуем два графика в одном окне

GOSUB ChangeYDF

Ymaxt = Ymax

Ymint = Ymin

GOSUB ChoiceYM

IF Ymaxt > Ymax THEN Ymax = Ymaxt

IF Ymint < Ymin THEN Ymin = Ymint

GOSUB CalcKBy

GOSUB Grid

COLOR 5

GOSUB Curve

GOSUB ChangeYF

' \
' |
' \ Автовыбор границ по Y
' /
' |
' /

' \ прорисовка графика производной

' /

```

COLOR 9
GOSUB Curve
END IF
IF (S$ = "N") OR (S$ = "n") THEN ' рисуем график только производной
GOSUB ChangeYDF ' \ Автовыбор границ по Y
GOSUB ChoiceYM ' /
GOSUB CalcKBy
GOSUB Grid
COLOR 5 ' \ прорисовка графика производной
GOSUB Curve ' /
END IF
COLOR 15
RETURN

```

MapY:

```

' Масштабирование кривой по оси Y.
COLOR 13
VIEW PRINT 20 TO 24: CLS 2
PRINT " Введите нижнюю и верхнюю границу по Y."
INPUT " Ymin="; Ymin
INPUT " Ymax="; Ymax
CLS 1 'очистка графического окна
GOSUB CalcKBy
GOSUB Grid
'GOSUB ChangeYF
COLOR 9 ' \ прорисовка графика в новом масштабе
GOSUB Curve ' /
COLOR 15
RETURN

```

' ГРАФИК

Curve:

```

FOR i = 0 TO 529 '100 TO 629
'Укладка графика в заданном интервале Ymin Ymax
X = (100 + i - Bx) / Kx
IF (Y(i) > Ymin) AND (Y(i) < Ymax) AND (Y(i + 1) > Ymin) AND (Y(i + 1) < Ymax) THEN
LINE (100 + i, Ky * Y(i) + By)-(101 + i, Ky * Y(i + 1) + By)
ELSE
'возможное соединение со следующей точкой
IF ((Ymin - Y(i)) * (Ymin - Y(i + 1)) < 0) THEN
Xp = CalcXp(X, Y(i), X + dx, Y(i + 1), Ymin)
IF (Y(i) < Ymin) THEN
LINE (Kx * Xp + Bx, Ky * Ymin + By)-(Kx * (X + dx) + Bx,

```

```

        Ky * Y(i + 1) + By)
    END IF
    IF (Y(i + 1) < Ymin) THEN
        LINE (Kx * Xp + Bx, Ky * Ymin + By)–(Kx * X + Bx, Ky * Y(i + 1) + By)
    END IF
END IF
IF ((Ymax – Y(i)) * (Ymax – Y(i + 1)) < 0) THEN
    Xp = CalcXp(X, Y(i), X + dx, Y(i + 1), Ymax)
    IF (Y(i) > Ymax) THEN
        LINE (Kx * Xp + Bx, Ky * Ymax + By)–(Kx * (X + dx) + Bx, Ky * Y(i + 1) + By)
    END IF
    IF (Y(i + 1) > Ymax) THEN
        LINE (Kx * Xp + Bx, Ky * Ymax + By)–(Kx * X + Bx, Ky * Y(i + 1) + By)
    END IF
END IF
NEXT i
COLOR 15
RETURN

```

ChoiceYM:

'ВЫБОР Ymin И Ymax для автоматического масштабирования по Y

Ymin = Y(0): Ymax = Y(0)

FOR i = 1 TO 530

IF Y(i) > Ymax THEN Ymax = Y(i)

IF Y(i) < Ymin THEN Ymin = Y(i)

NEXT i

IF Ymax = Ymin THEN

Ymax = Ymax + .1

Ymin = Ymin – .1

END IF

RETURN

ChangeYDF:

'замена массива Y(i) на DF(i)

FOR i = 0 TO 530

Y(i) = DF(i)

NEXT i

RETURN

ChangeYF:

'замена массива Y(i) на F(i)

FOR i = 0 TO 530

Y(i) = F(i)

NEXT i

RETURN

CalcKBy:

'расчет коэффициентов отображения по Y

Ky = 250 / (Ymin - Ymax): By = -Ky * Ymax

RETURN

Grid:

'РАЗМЕТКА

COLOR 15

VIEW PRINT

FOR i = 100 TO 630 STEP 53: LINE (i, 0)-(i, 250)

IF INT((i - 100) / 106) = (i - 100) / 106 THEN CIRCLE (i, 250), 3

NEXT i

FOR i = 0 TO 250 STEP 25: LINE (100, i)-(630, i): NEXT i

'НУЛЕВАЯ ЛИНИЯ

IF Ymax * Ymin ≤ 0 THEN LINE (100, By)-(630, By), 12

CALL REC(Ymax, 1, 1)

CALL REC(Ymin, 16, 1)

CALL REC((125 - By) / Ky, 8, 1)

M = 100

FOR i = 0 TO 5

CALL REC8((M - Bx) / Kx, 17, i * 13 + 6)

M = M + 106

NEXT i

RETURN

REM \$STATIC

FUNCTION CalcXp (X1, Y1, X2, Y2, Yc)

' вычисление абсциссы точки пересечения прямой, построенной

' на точках (X1;Y1) и (X2;Y2) с прямой Y=Yc

Kp = (Y2 - Y1) / (X2 - X1)

Bp = Y1 - Kp * X1

CalcXp = (Yc - Bp) / Kp

END FUNCTION

SUB REC (R AS SINGLE, PX, PY AS INTEGER)

LOCATE PX, PY

IF ABS(R) < 100 THEN PRINT USING "###.#####"; R: EXIT SUB

```
IF ABS(R) < 1000 THEN PRINT USING "####.#####"; R: EXIT SUB
IF ABS(R) < 100000 THEN PRINT USING "#####.#####"; R: EXIT SUB
IF ABS(R) < 10000000 THEN PRINT USING "#####.##"; R: EXIT SUB
IF ABS(R) > 9999999 THEN PRINT USING "#####"; R
END SUB
```

```
SUB REC8 (R AS SINGLE, PX, PY AS INTEGER)
LOCATE PX, PY
IF ABS(R) < 100 THEN PRINT USING "###.#####"; R: EXIT SUB
IF ABS(R) < 1000 THEN PRINT USING "####.#####"; R: EXIT SUB
IF ABS(R) < 100000 THEN PRINT USING "#####.###"; R: EXIT SUB
IF ABS(R) < 1000000 THEN PRINT USING "#####.##"; R: EXIT SUB
IF ABS(R) > 9999999 THEN PRINT USING "#####"; R
END SUB
```