**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**Лабораторная работа №3**

**по дисциплине «Организация ЭВМ и систем»**

**Тема: ИССЛЕДОВАНИЕ ВИДЕОСИСТЕМЫ (ГРАФИЧЕСКИЙ РЕЖИМ)**

| Студенты гр. 2302 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Николаев В.Ю. |
| --- | --- | --- |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Анисимов А.В. |

Санкт-Петербург

2023 г.

# Тестовые примеры запуска программы

|  |
| --- |

|  |
| --- |

# 

|  |
| --- |

# 

# Текст программы

| | 1.C #include <graphics.h>  #include <stdio.h>  #include <math.h>  static const double pi = 3.14159265358979323846;  int XX(double x, int indentX, double minX, double kX) {  return (int)((x - minX) / kX + indentX);  }  int YY(double y, int indentY, double maxY, double kY) {  return (int)((maxY - y) / kY + indentY);  }  int main() {  int gd = DETECT, gm;  int sizeX, sizeY;  int indentX = 40, indentY = 40;  double sx = pi / 2, ex = 8 \* pi, dx = pi / 180;  int size = (int)((ex - sx) / dx + 1);  double \*arrX = malloc(size \* sizeof(double));  double \*arrY = malloc(size \* sizeof(double));  char \*a = malloc(10 \* sizeof(char));  double minX = sx, maxX = ex;  double minY = 1e9, maxY = -1e9;  double kX, kY;  int i;  for (i = 0; i < size; i++) {  arrX[i] = sx + i \* dx;  arrY[i] = pow(sin(arrX[i]), 3) - pow(cos(arrX[i]), 2);  if (arrY[i] < minY) minY = arrY[i];  if (arrY[i] > maxY) maxY = arrY[i];  }  initgraph(&gd, &gm, "");  sizeX = getmaxx() - 2 \* indentX;  sizeY = getmaxy() - 2 \* indentY;  kX = (maxX - minX) / sizeX;  kY = (maxY - minY) / sizeY;  setcolor(WHITE);  for (i = 1; i < size; i++) {  line(XX(arrX[i - 1], indentX, minX, kX), YY(arrY[i - 1], indentY, maxY, kY), XX(arrX[i], indentX, minX, kX), YY(arrY[i], indentY, maxY, kY));  }  getch();  closegraph();  return 0;  } | | --- |  | 2.C #include <graphics.h>  #include <stdio.h>  #include <math.h>  static const double pi = 3.14159265358979323846;  int XX(double x, int indentX, double minX, double kX) {  return (int)((x - minX) / kX + indentX);  }  int YY(double y, int indentY, double maxY, double kY) {  return (int)((maxY - y) / kY + indentY);  }  int main() {  int gd = DETECT, gm;  int sizeX, sizeY;  int indentX = 40, indentY = 40;  double sx = pi / 2, ex = 8 \* pi, dx = pi / 180;  int size = (int)((ex - sx) / dx + 1);  double \*arrX = malloc(size \* sizeof(double));  double \*arrY = malloc(size \* sizeof(double));  char \*a = malloc(10 \* sizeof(char));  double minX = sx, maxX = ex;  double minY = 1e9, maxY = -1e9;  double kX, kY;  int i;  for (i = 0; i < size; i++) {  arrX[i] = sx + i \* dx;  arrY[i] = pow(sin(arrX[i]), 3) - pow(cos(arrX[i]), 2);  if (arrY[i] < minY) minY = arrY[i];  if (arrY[i] > maxY) maxY = arrY[i];  }  initgraph(&gd, &gm, "");  sizeX = getmaxx() - 2 \* indentX;  sizeY = getmaxy() - 2 \* indentY;  kX = (maxX - minX) / sizeX;  kY = (maxY - minY) / sizeY;  setcolor(DARKGRAY);  line(XX(minX, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY));  line(XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20 - 10, YY((maxY + minY) / 2, indentY, maxY, kY) - 5);  line(XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20 - 10, YY((maxY + minY) / 2, indentY, maxY, kY) + 5);  outtextxy(XX(maxX, indentX, minX, kX) + 20 - textwidth("X"), YY((maxY + minY) / 2, indentY, maxY, kY) + 10, "X");  for (i = (int)(sx + 1); i < ex; i++) {  itoa(i, a, 10);  outtextxy(XX(i, indentX, minX, kX) - textwidth(a) / 2 + 2, YY((maxY + minY) / 2, indentY, maxY, kY) + 10, a);  line(XX(i, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY) - 5, XX(i, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY) + 5);  }  line(XX(minX, indentX, minX, kX), YY(minY, indentY, maxY, kY) + 20, XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20);  line(XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20, XX(minX, indentX, minX, kX) - 5, YY(maxY, indentY, maxY, kY) - 20 + 10);  line(XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20, XX(minX, indentX, minX, kX) + 5, YY(maxY, indentY, maxY, kY) - 20 + 10);  outtextxy(XX(minX, indentX, minX, kX) - 10 - textwidth("Y"), YY(maxY, indentY, maxY, kY) - 20 + 2, "Y");  for (i = (int)(minY); i <= maxY; i++) {  itoa(i, a, 10);  outtextxy(XX(minX, indentX, minX, kX) - 10 - textwidth(a), YY(i, indentY, maxY, kY) - textheight(a) / 2, a);  line(XX(minX, indentX, minX, kX) - 5, YY(i, indentY, maxY, kY), XX(minX, indentX, minX, kX) + 5, YY(i, indentY, maxY, kY));  }  setcolor(WHITE);  for (i = 1; i < size; i++) {  line(XX(arrX[i - 1], indentX, minX, kX), YY(arrY[i - 1], indentY, maxY, kY), XX(arrX[i], indentX, minX, kX), YY(arrY[i], indentY, maxY, kY));  }  getch();  closegraph();  return 0;  } | | --- | | | 3.C #include <graphics.h>  #include <stdio.h>  #include <math.h>  static const double pi = 3.14159265358979323846;  int XX(double x, int indentX, double minX, double kX) {  return (int)((x - minX) / kX + indentX);  }  int YY(double y, int indentY, double maxY, double kY) {  return (int)((maxY - y) / kY + indentY);  }  double f(double x) {  return pow(sin(x), 3) - pow(cos(x), 2);  }  int main() {  int gd = DETECT, gm;  int sizeX, sizeY;  int indentX = 40, indentY = 40;  double sx = pi / 2, ex = 8 \* pi, dx = pi / 180;  int size = (int)((ex - sx) / dx + 1);  double \*arrX = malloc(size \* sizeof(double));  double \*arrY = malloc(size \* sizeof(double));  char \*a = malloc(10 \* sizeof(char));  double minX = sx, maxX = ex;  double minY = 1e9, maxY = -1e9;  double kX, kY;  int i;  for (i = 0; i < size; i++) {  arrX[i] = sx + i \* dx;  arrY[i] = f(arrX[i]);  if (arrY[i] < minY) minY = arrY[i];  if (arrY[i] > maxY) maxY = arrY[i];  }  initgraph(&gd, &gm, "");  sizeX = getmaxx() - 2 \* indentX;  sizeY = getmaxy() - 2 \* indentY;  kX = (maxX - minX) / sizeX;  kY = (maxY - minY) / sizeY;  setcolor(DARKGRAY);  line(XX(minX, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY));  line(XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20 - 10, YY((maxY + minY) / 2, indentY, maxY, kY) - 5);  line(XX(maxX, indentX, minX, kX) + 20, YY((maxY + minY) / 2, indentY, maxY, kY), XX(maxX, indentX, minX, kX) + 20 - 10, YY((maxY + minY) / 2, indentY, maxY, kY) + 5);  outtextxy(XX(maxX, indentX, minX, kX) + 20 - textwidth("X"), YY((maxY + minY) / 2, indentY, maxY, kY) + 10, "X");  for (i = (int)(sx + 1); i < ex; i++) {  itoa(i, a, 10);  outtextxy(XX(i, indentX, minX, kX) - textwidth(a) / 2 + 2, YY((maxY + minY) / 2, indentY, maxY, kY) + 10, a);  line(XX(i, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY) - 5, XX(i, indentX, minX, kX), YY((maxY + minY) / 2, indentY, maxY, kY) + 5);  }  line(XX(minX, indentX, minX, kX), YY(minY, indentY, maxY, kY) + 20, XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20);  line(XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20, XX(minX, indentX, minX, kX) - 5, YY(maxY, indentY, maxY, kY) - 20 + 10);  line(XX(minX, indentX, minX, kX), YY(maxY, indentY, maxY, kY) - 20, XX(minX, indentX, minX, kX) + 5, YY(maxY, indentY, maxY, kY) - 20 + 10);  outtextxy(XX(minX, indentX, minX, kX) - 10 - textwidth("Y"), YY(maxY, indentY, maxY, kY) - 20 + 2, "Y");  for (i = (int)(minY); i <= maxY; i++) {  itoa(i, a, 10);  outtextxy(XX(minX, indentX, minX, kX) - 10 - textwidth(a), YY(i, indentY, maxY, kY) - textheight(a) / 2, a);  line(XX(minX, indentX, minX, kX) - 5, YY(i, indentY, maxY, kY), XX(minX, indentX, minX, kX) + 5, YY(i, indentY, maxY, kY));  }  setcolor(WHITE);  for (i = 1; i < size; i++) {  line(XX(arrX[i - 1], indentX, minX, kX), YY(arrY[i - 1], indentY, maxY, kY), XX(arrX[i], indentX, minX, kX), YY(arrY[i], indentY, maxY, kY));  }  itoa((int)(minY), a, 10);  a[7] = a[0];  a[8] = a[1];  a[0] = 'm';  a[1] = 'i';  a[2] = 'n';  a[3] = 'Y';  a[4] = ' ';  a[5] = '=';  a[6] = ' ';  // a = "minX = " + a;  outtextxy(XX(maxX, indentX, minX, kX) - textwidth(a), YY(maxY, indentY, maxY, kY) + 10, a);  itoa((int)(maxY), a, 10);  a[7] = a[0];  a[8] = a[1];  a[0] = 'm';  a[1] = 'a';  a[2] = 'x';  a[3] = 'Y';  a[4] = ' ';  a[5] = '=';  a[6] = ' ';  // a = "maxX = " + a;  outtextxy(XX(maxX, indentX, minX, kX) - textwidth(a), YY(maxY, indentY, maxY, kY) + 10 + textheight(a) + 10, a);  getch();  closegraph();  return 0;  } | | --- | |
| --- | --- | --- | --- | --- |

# Вывод

В этой лабораторной работе мы изучили работу с видеосистемой в графическом режиме, вывели графика заданной функции с масштабированием и разметкой осей.