Задание 5

Вариант 12

a)

|  |
| --- |
| Площадь криволинейной трапеции , методом трапеций |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return exp(x) + 6;  }  double trap(double a, double b, int n) {  double h = (b - a) / n;  double x = a;  double s = 0.0;  while (x <= b - h) {  s += h \* (f(x) + f(x + h)) / 2;  x += h;  }  return s;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 5, b = 11;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции: " << area << endl;  return 0;  }  Результат программы |

б)

|  |
| --- |
| Площадь криволинейной трапеции , методом парабол |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return exp(x) + 6;  }  double trap(double a, double b, int n) {  double h = (b - a) / (2 \* n);  double x = a + 2 \* h;  double s1 = 0.0, s2 = 0.0;  for (int i = 1; i < n; i++) {  s2 += f(x);  x += h;  s1 += f(x);  x += h;  }  double S = (h / 3) \* (f(a) + 4 \* f(a + h) + 4 \* s1 + 2 \* s2 + f(b));  return S;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 5, b = 11;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции методом парабол :" << area << endl;  return 0;  }  Результат программы: |

Задание 6

Вариант 12

|  |
| --- |
| Написать программу вычисления корня уравнения методом ***дихотомии***. Точность вычислений принять равной **e** = 0,0001 для всех вариантов. (a=5, b=11) |

|  |
| --- |
| Графический метод:  Исходное уравнение:  x^3+2x−4=0  1. Разделение уравнения: Перепишем его в виде:  x^3+2x=4  2. Теперь у нас две функции:   * y1=x^3+2x - это левая часть уравнения. * y2=4 - это правая часть уравнения, просто горизонтальная линия на уровне y=4   3. Построив графики этих двух функций, мы видим , где они пересекаются. |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return (x \* x \* x) + (2 \* x) - 4;  }  double dichom(double a, double b, double e) {  double x;  do  {  x = (a + b) / 2;  if ((f(x) \* f(a)) <= 0)  b = x;  else  a = x;  } while (abs(a - b) >= 2 \* e);  return x;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 1, b = 2, e = 0.0001;  double area = dichom(a, b, e);  cout << "X= " << area << endl;  return 0;  }  Результат программы:    Проверка с помощью Exсel:  A1=1    A2=2 |

Доп.задания:

Вариант (12+5=17=1)

№5

a)

|  |
| --- |
| Площадь криволинейной трапеции , методом трапеций |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x\*x\*x-3;  }  double trap(double a, double b, int n) {  double h = (b – a) / n;  double x = a;  double s = 0.0;  while (x <= b – h) {  s += h \* (f(x) + f(x + h)) / 2;  x += h;  }  return s;  }  int main() {  setlocale(LC\_ALL, “ru”);  double a = 1, b = 3;  int n = 200;  double area = trap(a, b, n);  cout << «Площадь криволинейной трапеции: « << area << endl;  return 0;  }  Результат программы: |

б)

|  |
| --- |
| Площадь криволинейной трапеции , методом парабол |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x\*x\*x-3;  }  double trap(double a, double b, int n) {  double h = (b – a) / (2 \* n);  double x = a + 2 \* h;  double s1 = 0.0, s2 = 0.0;  for (int I = 1; I < n; i++) {  s2 += f(x);  x += h;  s1 += f(x);  x += h;  }  double S = (h / 3) \* (f(a) + 4 \* f(a + h) + 4 \* s1 + 2 \* s2 + f(b));  return S;  }  int main() {  setlocale(LC\_ALL, “ru”);  double a = 1, b = 3;  int n = 200;  double area = trap(a, b, n);  cout << Площадь криволинейной трапеции методом парабол : « << area << endl;  return 0;  }  Результат программы: |

Задание 6

|  |
| --- |
| Исходное уравнение:  x^3+x−3=0  1. Разделение уравнения: Перепишем уравнение в виде:  x3+x=3  2. Теперь у нас две функции:   * y1=x^3+x — это левая часть уравнения. * y2=3 — это правая часть уравнения, которая представляет собой горизонтальную линию на уровне y=3.   3. Построив графики этих двух функций, мы видим , где они пересекаются.    #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x \* x \* x + x – 3;  }  double dichom(double a, double b, double e) {  double x;  do  {  x = (a + b) / 2;  if ((f(x) \* f(a)) <= 0)  b = x;  else  a = x;  } while (abs(a – b) >= 2 \* e);  return x;  }  int main() {  setlocale(LC\_ALL, “ru”);  double a = 1, b = 2, e = 0.0001;  double area = dichom(a, b, e);  cout << “X= “ << area << endl;  return 0;  }  Результат программы:    Проверка с помощью Exсel:  A1=1 |

Вариант(12+2=14)

Задание 5

a)

|  |
| --- |
| Площадь криволинейной трапеции , методом трапеций |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x \* x \* x\*x+4;  }  double trap(double a, double b, int n) {  double h = (b-a) / n;  double x = a;  double s = 0.0;  while (x <= (b-h)) {  s += h \* (f(x) + f(x + h)) / 2;  x += h;  }  return s;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 1, b = 4;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции :" << area << endl;  return 0;  }  Результат программы: |

б)

|  |
| --- |
| Площадь криволинейной трапеции , методом парабол |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x\*x\*x\*x+4;  }  double trap(double a, double b, int n) {  double h = (b - a) / (2 \* n);  double x = a + 2 \* h;  double s1 = 0.0, s2 = 0.0;  for (int i = 1;i < n; i++) {  s2 += f(x);  x += h;  s1 += f(x);  x += h;  }  double S = (h / 3) \* (f(a) + 4 \* f(a + h) + 4 \* s1 + 2 \* s2 + f(b));  return S;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 1, b = 4;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции методом парабол :" << area << endl;  return 0;  }  Результат программы: |

Задание 6

|  |
| --- |
| Исходное уравнение:  x^3+x−2=0  1. Разделение уравнения: Перепишем уравнение в виде:  x3+x=2  2. Теперь у нас две функции:   * y1=x^3+x — это левая часть уравнения. * y2=2 — это правая часть уравнения, которая представляет собой горизонтальную линию на уровне y=2.   3. Построив графики этих двух функций, мы видим , где они пересекаются.    #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return x \* x \* x + x - 2;  }  double dichom(double a, double b, double e) {  double x;  do  {  x = (a + b) / 2;  if ((f(x) \* f(a)) <= 0)  b = x;  else  a = x;  } while (abs(a - b) >= 2 \* e);  return x;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 0, b = 2, e = 0.0001;  double area = dichom(a, b, e);  cout << "X = " << area << endl;  return 0;  }  Результат программы:    Проверка с помощью Exсel:  A1=1 |

Вариант (12-6=6)

Задание 5

a)

|  |
| --- |
| Площадь криволинейной трапеции , методом трапеций |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return 1+x \* x \* x;  }  double trap(double a, double b, int n) {  double h = (b - a) / n;  double x = a;  double s = 0.0;  while (x <= (b - h)) {  s += h \* (f(x) + f(x + h)) / 2;  x += h;  }  return s;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 1, b = 5;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции :" << area << endl;  return 0;  }  Результат программы: |

б)

|  |
| --- |
| Площадь криволинейной трапеции , методом парабол |

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return 1+x \* x \* x;  }  double trap(double a, double b, int n) {  double h = (b - a) / (2 \* n);  double x = a + 2 \* h;  double s1 = 0.0, s2 = 0.0;  for (int i = 1; i < n; i++) {  s2 += f(x);  x += h;  s1 += f(x);  x += h;  }  double S = (h / 3) \* (f(a) + 4 \* f(a + h) + 4 \* s1 + 2 \* s2 + f(b));  return S;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 1, b = 5;  int n = 200;  double area = trap(a, b, n);  cout << "Площадь криволинейной трапеции методом парабол :" << area << endl;  return 0;  }  Результат программы: |

Задание 6

|  |
| --- |
| Исходное уравнение:  5x−1+x^3=0  1. Разделение уравнения: Перепишем уравнение в виде:  5x+x^3=1  2. Теперь у нас две функции:   * y1=x^3+5x — это левая часть уравнения. * y2=1 — это правая часть уравнения, которая представляет собой горизонтальную линию на уровне y=1.   3. Построив графики этих двух функций, мы видим , где они пересекаются.    #include <iostream>  #include <cmath>  using namespace std;  double f(double x) {  return 5 \* x - 1 + x \* x \* x;  }  double dichom(double a, double b, double e) {  double x;  do  {  x = (a + b) / 2;  if ((f(x) \* f(a)) <= 0)  b = x;  else  a = x;  } while (abs(a - b) >= 2 \* e);  return x;  }  int main() {  setlocale(LC\_ALL, "ru");  double a = 0, b = 1, e = 0.0001;  double area = dichom(a, b, e);  cout << "X = " << area << endl;  return 0;  }  Результат программы:    Проверка с помощью Exсel:  A1=1 |