Варыянт 3

Заданне 5 В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом трапеций и методом парабол. Для всех вариантов принять n = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |
| --- | --- |
| 1 +x3 | a = 1, b = 6 |

|  |
| --- |
| #include <iostream>  using namespace std;  // Метад трапецыі  int main()  {  int a = 1, b = 6, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  double h = (b - a);  h = h / n;  // Перабіраем усе лікі ад 0 да n+1  for (int i = 0; i <= n; i++) {  // Знаходзім х: для i = 0 будзе х0, які не улічваецца у формуле, але патрэбны для y0  xi = a + h \* i;  // Вылічваем у  y = 1 + pow(xi, 3);  // Для усіх yi, акрамя у0 і уn, дабаўляем іх падвоенае значэнне да зменнай presum, інакш дабаўляем сам y  presum += (i != 0 && i != n) ? 2 \* y : y;  }  // Знаходзім плошчу  S = h / 2 \* presum;  // Выводзім вынік  cout << S;  } |
|  |
| #include <iostream>  using namespace std;  // Метад парабалы  int main()  {  int a = 1, b = 6, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  float h = (b - a);  h = h / (2 \* n);  // Перабіраем усе лікі ад 0 да 2n  for (int i = 0; i <= 2 \* n; i++) {  xi = a + h \* i;  y = 1 + pow(xi, 3); // Вылічваем у  // Для усіх i, акрамя 0 і 2n: калі i - цотны, то павялічваем суму на 4у, інакш на 2у. калі і = 0 або і = 2n, то проста дадаем да сумы у  presum += (i != 0 && i != 2 \* n && i % 2 != 0) ? 4 \* y : (i != 0 && i != 2 \* n && i % 2 == 0) ? 2 \* y : y;  }  S = h / 3 \* presum;  cout << S;  } |
|  |

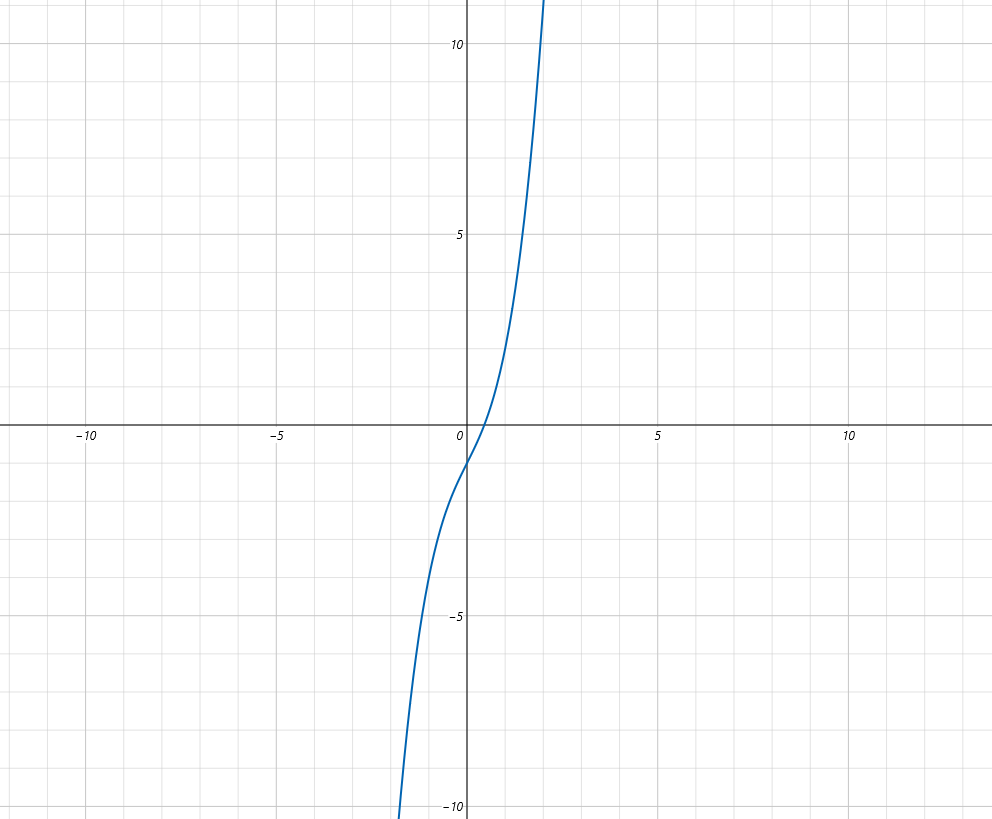
Заданне 6. В соответствии со своим вариантом найти отрезок (значения a и b), который содержит один корень, отделив кор-ни уравнения графическим методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

Написать программу вычисления корня уравнения методом дихотомии. Точность вычислений принять равной e = 0,0001 для всех вариантов.

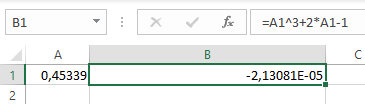
Найти корень уравнения с помощью приложения Excel.

Сравнить результаты.

x3 + 2x – 1

кропкі a b узяў з графіку функцыіі 

|  |
| --- |
| #include <iostream>  using namespace std;  //Функцыя для вылічэння значэння дадзенай функцыі у пукце х  double func(double x) {  return (pow(x, 3) + 2 \* x - 1);  }  int main()  {  double a = -5, b = 5, e = 0.0001;  double x;  x = (a + b) / 2; // Вылічваем х  // Пакуль умова выконваецца - працягваем пераўтварэнні  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (func(x) \* func(a) <= 0) {  b = x;  }  else {  a = x;  }  }  // Выводзім вынік  cout << x;  } |
|  |



Варыянт 1

Заданне 5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом трапеций и методом парабол. Для всех вариантов принять n = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |
| --- | --- |
| x3 – 3 | a = 1, b = 3 |

|  |
| --- |
| #include <iostream>  using namespace std;  // Метад трапецыі  int main()  {  int a = 1, b = 3, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  double h = (b - a);  h = h / n;  // Перабіраем усе лікі ад 0 да n  for (int i = 0; i <= n; i++) {  // Знаходзім х: для i = 0 будзе х0, які не улічваецца у формуле, але патрэбны для y0  xi = a + h \* i;  // Вылічваем у  y = pow(xi, 3) - 3;  // Для усіх yi, акрамя у0 і уn, дабаўляем іх падвоенае значэнне да зменнай presum, інакш дабаўляем сам y  presum += (i != 0 && i != n) ? 2 \* y : y;  }  // Знаходзім плошчу  S = h / 2 \* presum;  // Выводзім вынік  cout << S;  } |
|  |

|  |
| --- |
| #include <iostream>  using namespace std;  // Метад парабалы  int main()  {  int a = 1, b = 3, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  float h = (b - a);  h = h / (2 \* n);  // Перабіраем усе лікі ад 0 да 2n  for (int i = 0; i <= 2 \* n; i++) {  xi = a + h \* i;  y = pow(xi, 3) - 3; // Вылічваем у  // Для усіх i, акрамя 0 і 2n: калі i - цотны, то павялічваем суму на 4у, інакш на 2у. калі і = 0 або і = 2n, то проста дадаем да сумы у  presum += (i != 0 && i != 2 \* n && i % 2 != 0) ? 4 \* y : (i != 0 && i != 2 \* n && i % 2 == 0) ? 2 \* y : y;  }  S = h / 3 \* presum;  cout << S;  } |
|  |

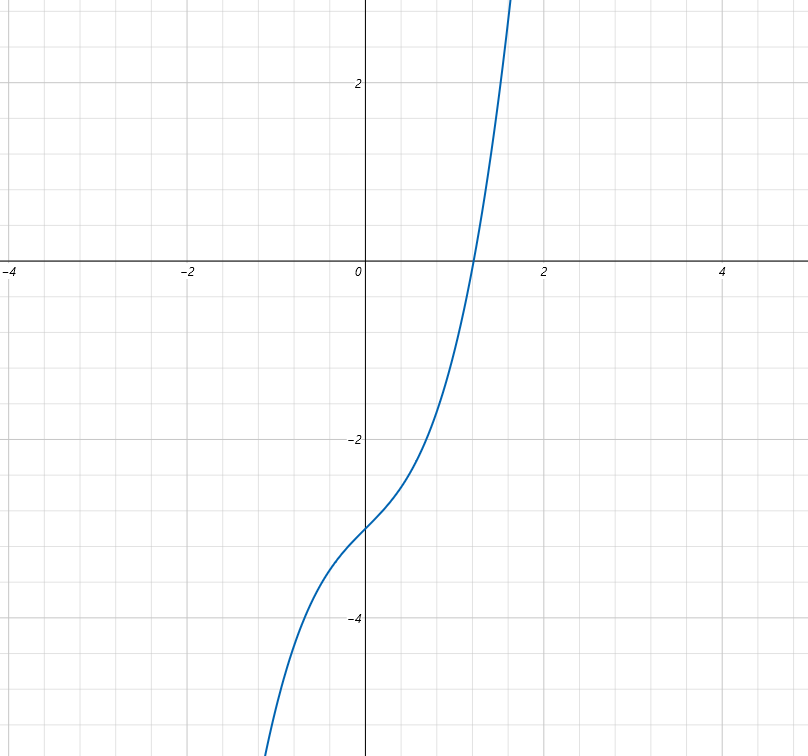
Зданне 6. В соответствии со своим вариантом найти отрезок (значения a и b), который содержит один корень, отделив кор-ни уравнения графическим методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

Написать программу вычисления корня уравнения методом дихотомии. Точность вычислений принять равной e = 0,0001 для всех вариантов.

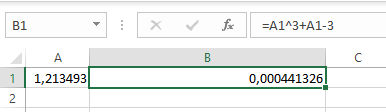
Найти корень уравнения с помощью приложения Excel.

Сравнить результаты.

x3 + x – 3



|  |
| --- |
| #include <iostream>  using namespace std;  //Функцыя для вылічэння значэння дадзенай функцыі у пукце х  double func(double x) {  return (pow(x, 3) + x - 3);  }  int main()  {  double a = -2, b = 2, e = 0.0001;  double x;  x = (a + b) / 2; // Вылічваем х  // Пакуль умова выконваецца - працягваем пераўтварэнні  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (func(x) \* func(a) <= 0) {  b = x;  }  else {  a = x;  }  }  // Выводзім вынік  cout << x;  } |
|  |



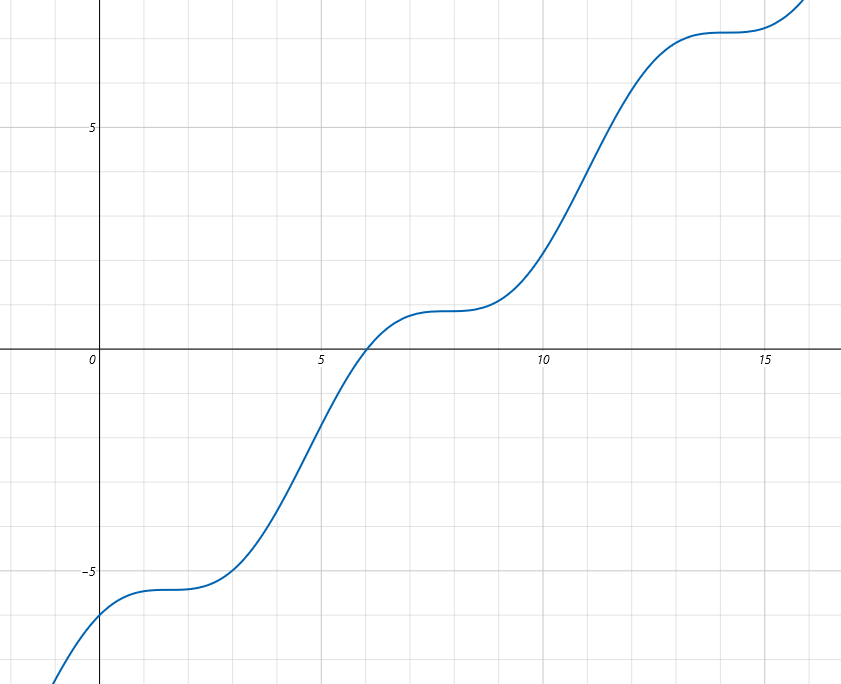
Варыянт 2

Заданне 5

|  |
| --- |
| #include <iostream>  using namespace std;  // Метад трапецыі  int main()  {  int a = 4, b = 7, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  double h = (b - a);  h = h / n;  // Перабіраем усе лікі ад 0 да n+1  for (int i = 0; i <= n; i++) {  // Знаходзім х: для i = 0 будзе х0, які не улічваецца у формуле, але патрэбны для y0  xi = a + h \* i;  // Вылічваем у  y = pow(cos(xi), 3);  // Для усіх yi, акрамя у0 і уn, дабаўляем іх падвоенае значэнне да зменнай presum, інакш дабаўляем сам y  presum += (i != 0 && i != n) ? 2 \* y : y;  }  // Знаходзім плошчу  S = h / 2 \* presum;  // Выводзім вынік  cout << S;  } |
|  |
| #include <iostream>  using namespace std;  // Метад парабалы  int main()  {  int a = 4, b = 7, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  float h = (b - a);  h = h / (2 \* n);  // Перабіраем усе лікі ад 0 да 2n  for (int i = 0; i <= 2 \* n; i++) {  xi = a + h \* i;  y = pow(cos(xi), 3); // Вылічваем у  // Для усіх i, акрамя 0 і 2n: калі i - цотны, то павялічваем суму на 4у, інакш на 2у. калі і = 0 або і = 2n, то проста дадаем да сумы у  presum += (i != 0 && i != 2 \* n && i % 2 != 0) ? 4 \* y : (i != 0 && i != 2 \* n && i % 2 == 0) ? 2 \* y : y;  } |
|  |

Заданне 6. В соответствии со своим вариантом найти отрезок (значения a и b), который содержит один корень, отделив кор-ни уравнения графическим методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

cos(x) + x – 7



|  |
| --- |
| #include <iostream>  using namespace std;  //Функцыя для вылічэння значэння дадзенай функцыі у пукце х  double func(double x) {  return (cos(x) + x - 7);  }  int main()  {  double a = 0, b = 10, e = 0.0001;  double x;  x = (a + b) / 2; // Вылічваем х  // Пакуль умова выконваецца - працягваем пераўтварэнні  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (func(x) \* func(a) <= 0) {  b = x;  }  else {  a = x;  }  }  // Выводзім вынік  cout << x;  } |
|  |
|  |

Варыянт 6

Заданне 5 В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исход-ным данным из таблицы, приведенной ниже, методом трапеций и методом парабол. Для всех вариантов принять n = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |
| --- | --- |
| 1 + x3 | a = 1, b = 5 |

|  |
| --- |
| #include <iostream>  using namespace std;  // Метад трапецыі  int main()  {  int a = 1, b = 5, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  double h = (b - a);  h = h / n;  // Перабіраем усе лікі ад 0 да n+1  for (int i = 0; i <= n; i++) {  // Знаходзім х: для i = 0 будзе х0, які не улічваецца у формуле, але патрэбны для y0  xi = a + h \* i;  // Вылічваем у  y = 1 + pow(xi, 3);  // Для усіх yi, акрамя у0 і уn, дабаўляем іх падвоенае значэнне да зменнай presum, інакш дабаўляем сам y  presum += (i != 0 && i != n) ? 2 \* y : y;  }  // Знаходзім плошчу  S = h / 2 \* presum;  // Выводзім вынік  cout << S;  } |
|  |
| #include <iostream>  using namespace std;  // Метад парабалы  int main()  {  int a = 1, b = 5, n = 200;  double y, S, presum = 0, xi;  // Вылічваем вышыню  float h = (b - a);  h = h / (2 \* n);  // Перабіраем усе лікі ад 0 да 2n  for (int i = 0; i <= 2 \* n; i++) {  xi = a + h \* i;  y = 1 + pow(xi, 3); // Вылічваем у  // Для усіх i, акрамя 0 і 2n: калі i - цотны, то павялічваем суму на 4у, інакш на 2у. калі і = 0 або і = 2n, то проста дадаем да сумы у  presum += (i != 0 && i != 2 \* n && i % 2 != 0) ? 4 \* y : (i != 0 && i != 2 \* n && i % 2 == 0) ? 2 \* y : y;  }  S = h / 3 \* presum;  cout << S;  } |
|  |

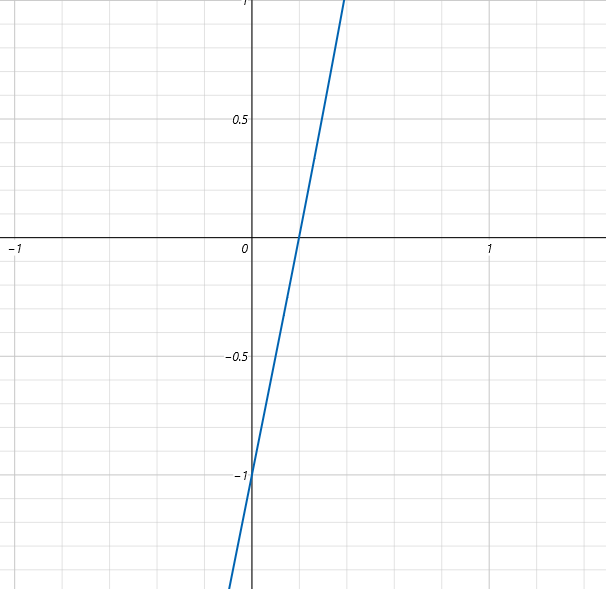
Заданне 6. В соответствии со своим вариантом найти отрезок (значения a и b), который содержит один корень, отделив корни уравнения графическим методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то вы-брать один из отрезков.

Написать программу вычисления корня уравнения методом дихотомии. Точность вычислений принять равной e = 0,0001 для всех вариантов.

Найти корень уравнения с помощью приложения Excel.

Сравнить результаты.

5x – 1 + x3



|  |
| --- |
| #include <iostream>  using namespace std;  //Функцыя для вылічэння значэння дадзенай функцыі у пукце х  double func(double x) {  return (pow(x, 3) + 5 \* x - 1);  }  int main()  {  double a = 0, b = 1, e = 0.0001;  double x;  x = (a + b) / 2; // Вылічваем х  // Пакуль умова выконваецца - працягваем пераўтварэнні  while (abs(a - b) > 2 \* e) {  x = (a + b) / 2;  if (func(x) \* func(a) <= 0) {  b = x;  }  else {  a = x;  }  }  // Выводзім вынік  cout << x;  } |
|  |
|  |