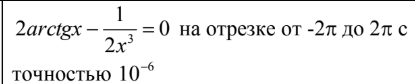
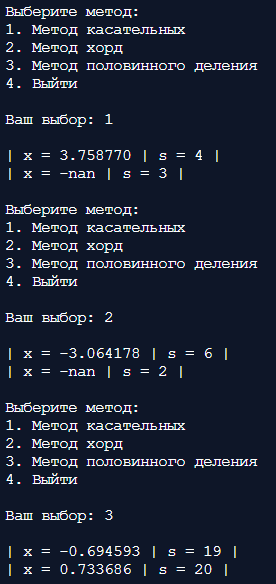
**Лабораторная работа №5. Нелинейные уравнения.**

1. Код лабораторной работы написан на языке: С;

Компилятор: онлайн среда Repl.it (clang version 7.0.0)

1. Постановка задачи: Вычислить значение нелинейного уравнения:



1. Результат:  
   
2. Код: https://repl.it/@sonyadk/ny

| Код |
| --- |
| /\*Решить нелинейные уравнения методами деления отрезка пополам, касательных, хорд:  y = x^3 - 12\*x - 8 = 0, e=10^-4, [-3,4]  y = 2\*arctng(x)-1/2\*x^3 = 0, [-2pi, 2pi]  \*/  #include <stdio.h>  #include <math.h>  #include <stdlib.h>  #define M\_PI 3.14159265358979323846 /\* pi \*/  float a\_leck=-3;  float b\_leck=4;  float a\_lich=-2\*M\_PI;  float b\_lich=2\*M\_PI;  float e=0.00001;  float leck(float x); //общее уравнение  float pr\_leck(float x); //производная общего уравнения  float lich(float x); //индивидуальное уравнение  float pr\_lich(float x);//производная индивидуального уравнения  void hord(); //метод хорд  void half(); //метод половинного деления  void tang(); //метод касательных  void Menu();  void Menu(){  printf("\nВыберите метод: \n");  printf("1. Метод касательных \n");  printf("2. Метод хорд \n");  printf("3. Метод половинного деления \n");  printf("4. Выйти \n");  int num;  printf("\nВаш выбор: ");  scanf("%d", &num);  switch (num){  case 1:  tang();  break;  case 2:  hord();  break;  case 3:  half();  break;  case 4:  exit(0);  }  }  float leck(float x){  float y = pow(x,3) - 12\*x - 8;  return y;  }  float pr\_leck(float x){  float z = (3\*x\*x-12)\*6\*x;  return z;  }  float lich(float x){  float m = 2\*atan(x) - 1/(2\*pow(x,3));  return m;  }  float pr\_lich(float x){  float v = 3/(2\*pow(x,4)) + 2/(1 + x\*x);  return v;  }  void half(){  float a = a\_leck, b = b\_leck, c=0,x=0;  int s=0;  do{  x=(a+b)/2;  if(leck(a)\*leck(x)<0) {  b = x;}  else {  a=x; }  s++;  }while (b-a>2\*e);  printf("\n| x = %f | s = %d |", x, s);  a = a\_lich, b = b\_lich, c=0,x=0;  s=0;  do{  x=(a+b)/2;  if(lich(a)\*lich(x)<0) {  b = x;}  else {  a=x; }  s++;  }while (b-a>2\*e);  printf("\n| x = %f | s = %d |\n", x, s);  Menu();  }  void tang(){  float c=0,x=0,x0=0;  int s=0;  if (leck(a\_leck)\*pr\_leck(a\_leck)>0){  x0=a\_leck;}  else {  x0=b\_leck;}  c=1;  while (c>e){  x=x0-leck(x0)\*e/(leck(x0+e)-leck(x0));  c=fabs(x-x0);  x0=x;  s++;  }  printf("\n| x = %f | s = %d |", x, s);  c=0,x=0,x0=0;  s=0;  if (lich(a\_lich)\*pr\_lich(a\_lich)>0){  x0=a\_lich;}  else {  x0=b\_lich;}  c=1;  while (c>e){  x=x0-lich(x0)\*e/(lich(x0+e)-lich(x0));  c=fabs(x-x0);  x0=x;  s++;  }  printf("\n| x = %f | s = %d |\n", x, s);  Menu();  }  void hord(){  float x = 0, x0 = 0, c;  int s;  if (pr\_leck(a\_leck) > 0) {  x = a\_leck;  c = b\_leck;  }  else {  x = b\_leck;  c = a\_leck;  }  s = 0;  do{  x0 = x;  x = x0 - leck(x0)\*(c-x0)/(leck(c)-leck(x0));  s++;  }while(fabs(x0-x)>e);  printf("\n| x = %f | s = %d |", x, s);  x0 = 0; x = 0;  c = 0;  if (pr\_lich(a\_lich) > 0) {  x = a\_lich;  c = b\_lich;  }  else {  x = b\_lich;  c = a\_lich;  }  s = 0;  do{  x0 = x;  x = x0 - lich(x0)\*(c - x0)/(lich(c)-lich(x0));  s++;  }while(fabs(x-x0)>e);  printf("\n| x = %f | s = %d |\n", x, s);  Menu();  }  int main(void) {  Menu();  return 0;  } |