**Работа над ошибками**

**Лабораторная работа N1**

**Вариант:18.01.1998**

**Распечатка файла even-odd.cpp**

**#include "mlisp.h"**

**double dd = 18;**

**double mm = 01;**

**double yyyy = 1998;**

**double even\_\_bits(double);**

**double odd\_\_bits(double);**

**double odd\_\_bits(double n)**

**{**

**return (n == 0 ? 0**

**: remainder(n,2) == 0 ?**

**odd\_\_bits (quotient(n,2))**

**: true ? even\_\_bits(quotient(n,2))**

**: \_infinity);**

**}**

**double even\_\_bits(double n)**

**{**

**return (n == 0 ? 1**

**: remainder(n,2) == 0 ?**

**even\_\_bits (quotient(n,2))**

**: (odd\_\_bits(quotient(n,2))));**

**}**

**double display\_\_bin(double n)**

**{**

**display(remainder(n,2));**

**return (n == 0 ? 0**

**: display\_\_bin(quotient(n,2)));**

**}**

**double report\_\_results(double n)**

**{**

**display("Happy birthday to you!\n\t");**

**display(n);**

**display("(decimal)\n\t");**

**display\_\_bin(n);**

**display("(reversed binary)\n");**

**display("\teven?\t");**

**display((even\_\_bits(n) == 1 ? "yes" : "no"));**

**newline();**

**display("\todd?\t");**

**display((odd\_\_bits(n) == 1 ? "yes" : "no"));**

**newline();**

**return 0;**

**}**

**int main(){**

**display(report\_\_results (dd\*1000000+**

**mm\*10000+**

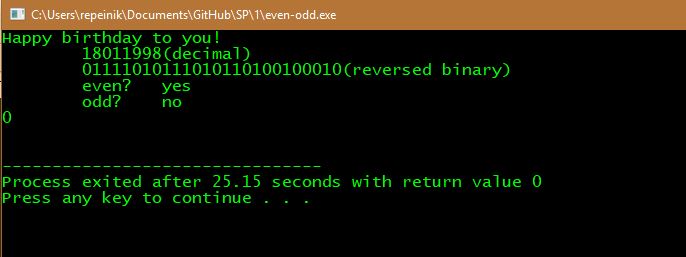
**yyyy));**

**newline();**

**std::cin.get();**

**}**

**Скриншот запуска на С++**



**Лабораторная работа N2**

**Вариант:06**

**Распечатка файла half-interval.cpp**

**#include "mlisp.h"**

**double tolerance = 0.00001;**

**double fun(double z);**

**double half\_\_interval\_\_metod(double a, double b);**

**double \_\_BAT\_\_try(double neg\_\_point, double pos\_\_point);**

**double average(double a, double b);**

**bool close\_\_enough\_Q(double neg\_\_point, double pos\_\_point);**

**double root(double a, double b);**

**double fun(double z)**

**{**

**z = z - (double)106/107 - (double)1/e;**

**return 0.25 \* expt(z, 3) + z - 1.2502;**

**}**

**double average(double a, double b)**

**{**

**return (double)(a + b) / 2;**

**}**

**bool close\_\_enough\_Q(double x, double y)**

**{**

**abs(x - y) < tolerance ? true : false;**

**}**

**double half\_\_interval\_\_metod(double a, double b)**

**{**

**{**

**double a\_\_value = fun(a);**

**double b\_\_value = fun(b);**

**return ((a\_\_value < 0 && b\_\_value > 0) ? \_\_BAT\_\_try(a, b)**

**: (a\_\_value > 0 && b\_\_value < 0) ? \_\_BAT\_\_try(b, a)**

**: b = b + 1);**

**}**

**}**

**double \_\_BAT\_\_try(double neg\_\_point, double pos\_\_point)**

**{**

**{**

**double midpoint = average(neg\_\_point, pos\_\_point);**

**double test\_\_value = 0;**

**display("+");**

**return(close\_\_enough\_Q(neg\_\_point, pos\_\_point) ? midpoint**

**: test\_\_value = fun(midpoint), test\_\_value > 0 ? \_\_BAT\_\_try(neg\_\_point, midpoint)**

**: test\_\_value < 0 ? \_\_BAT\_\_try(midpoint, pos\_\_point)**

**: midpoint);**

**}**

**}**

**double root(double a, double b)**

**{**

**display(" BAT variant 6");newline();**

**display("interval=\t[");**

**display(a);**

**display(" , ");**

**display(b);**

**display("]\n");**

**{**

**double temp = half\_\_interval\_\_metod(a, b);**

**newline();**

**display("discrepancy=\t");**

**display(fun(temp));**

**newline();**

**display("root=\t\t");**

**display(((temp - b - 1) == 0) ? "[bad]" : "[good]");**

**return temp;**

**}**

**}**

**int main()**

**{**

**display(root(2,3));**

**std::cin.get();**

**}**

**Скриншот запуска на C++**

