1. В программе из ЛР № 1 выделить код для преобразования температуры между шкалами в функцию.

*double convert(double t, char from, char to)*

*{ double K;*

*if (to==from) return t;*

*switch (from)*

*{*

*case 'K':*

*case 'k': K=t;*

*break;*

*case 'C':*

*case 'c': K=t+273.15;*

*break;*

*case 'F':*

*case 'f': K=5/9\*(t-32)+273.15;*

*break;*

*default: cout <<"Incorrect nomination of scale! Try again! \n";*

*}*

*if (K<0)*

*{cout <<"Too low temperature!\n";*

*return 0;*

*}*

*if (K>=0)*

*{ switch (to)*

*{*

*case 'C':*

*case 'c': return (K-273.15);*

*case 'F':*

*case 'f': return (9/5\*(K-273.15)+32);*

*case 'K':*

*case 'k': return K;*

*default: cout <<"Incorrect nomination of scale! Try again! \n";*

*}*

*}*

*}*

2. Перечисление кодов ошибок, которые могут возникать в функции convert()

|  |  |
| --- | --- |
| Код ошибки | Ошибка |
| 1 | Неизвестная шкала from или to |
| 2 | Слишком низкая температура (< 0K) |

1. Реализовать обработку ошибок вовне convert()по кодам возврата.

*#include "sdt.h"*

*int convert(double t, char from, char to)*

*{ double K,t1;*

*if (to==from) return t;*

*switch (from)*

*{*

*case 'k': K=t;*

*break;*

*case 'K': K=t;*

*break;*

*case 'C': K=t+273.15;*

*break;*

*case 'c': K=t+273.15;*

*break;*

*case 'F': K=(5\*(t-32))/9+273.15;*

*break;*

*case 'f': K=(5\*(t-32))/9+273.15;*

*break;*

*default: return 1;*

*}*

*if (K<0) return 2;*

*if (K>=0)*

*{ switch (to)*

*{*

*case 'C': return (K-273.15); break;*

*case 'c': return (K-273.15); break;*

*case 'F': return ((9\*(K-273.15))/5+32); break;*

*case 'f': return ((9\*(K-273.15))/5+32); break;*

*case 'K': return K; break;*

*case 'k': return K; break;*

*default: return 1;*

*}*

*}*

*}*

*int main()*

*{*

*double t;*

*char s;*

*vector<double> temp;*

*vector<char> name;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*while (cin>>t>>s)*

*{*

*switch (convert(t,s,'K'))*

*{ case 1: cout <<"Incorrect nomination of scale! Try again! \n";*

*break;*

*case 2: cout <<"Too low temperature!\n";*

*break;*

*default:*

*temp.push\_back(convert(t,s,'C'));*

*temp.push\_back(convert(t,s,'K'));*

*temp.push\_back(convert(t,s,'F'));*

*}*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*}*

*cout <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < temp.size(); ++i)*

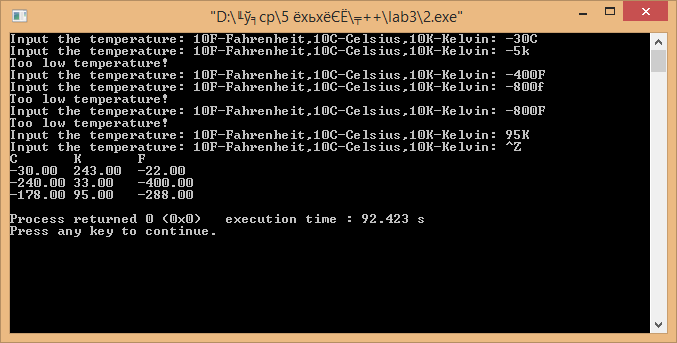
*{*

*printf("%4.2f\t", temp[i]);*

*if ((i+1)%3==0) cout <<"\n";*

*}*

*}*



1. Реализовать обработку ошибок вовне convert() через код последней ошибки.

*#include "sdt.h"*

*int last\_error = 0;*

*int get\_last\_error()*

*{*

*return last\_error;*

*}*

*double convert(double t, char from, char to)*

*{ double K,t1;*

*if (to==from) return t;*

*switch (from)*

*{*

*case 'k': K=t; break;*

*case 'K': K=t; break;*

*case 'C': K=t+273.15; break;*

*case 'c': K=t+273.15; break;*

*case 'F': K=(5\*(t-32))/9+273.15; break;*

*case 'f': K=(5\*(t-32))/9+273.15; break;*

*default: last\_error=1;*

*}*

*if (K<0) last\_error=2;*

*if (K>=0)*

*{ switch (to)*

*{*

*case 'C': return (K-273.15); break;*

*case 'c': return (K-273.15); break;*

*case 'F': return ((9\*(K-273.15))/5+32); break;*

*case 'f': return ((9\*(K-273.15))/5+32); break;*

*case 'K': return K; break;*

*case 'k': return K; break;*

*default: last\_error=1;*

*}*

*}*

*}*

*int main()*

*{*

*double t,C,K,F;*

*char s;*

*vector<double> temp;*

*vector<char> name;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*while (cin>>t>>s)*

*{ C=convert(t,s,'C');*

*F=convert(t,s,'K');*

*K=convert(t,s,'F);*

*switch (get\_last\_error())*

*{ case 1: cerr <<"Incorrect nomination of scale! Try again! \n";*

*break;*

*case 2: cerr <<"Too low temperature!\n";*

*break;*

*default:*

*temp.push\_back(C);*

*temp.push\_back(K);*

*temp.push\_back(F);*

*}*

*last\_error = 0;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*}*

*cout <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < temp.size(); ++i)*

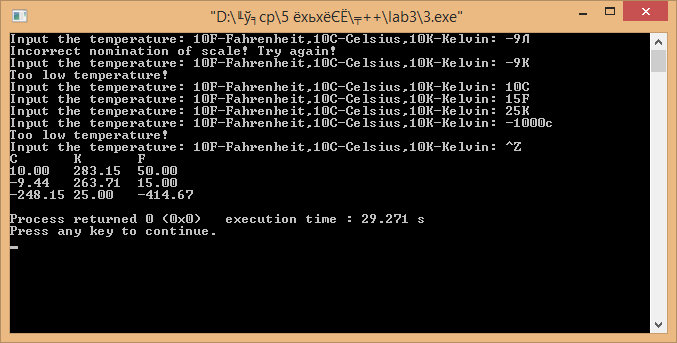
*{*

*printf("%4.2f\t", temp[i]);*

*if ((i+1)%3==0) cout <<"\n";*

*}*

*}*



5. Реализовать обработку ошибок вовне функции через механизм исключительных ситуаций.

*#include "sdt.h"*

*double convert(double t, char from, char to)*

*{ double K,t1;*

*if (to==from) return t;*

*switch (from)*

*{*

*case 'k':*

*case 'K': K=t;*

*break;*

*case 'C':*

*case 'c': K=t+273.15;*

*break;*

*case 'F':*

*case 'f': K=(5\*(t-32))/9+273.15;*

*break;*

*default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*if (K<0) throw logic\_error("Too low temperature!\n");*

*if (K>=0)*

*{ switch (to)*

*{*

*case 'C':*

*case 'c': return (K-273.15); break;*

*case 'F':*

*case 'f': return ((9\*(K-273.15))/5+32); break;*

*case 'K':*

*case 'k': return K; break;*

*default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*}*

*throw exception();*

*}*

*int main()*

*{*

*double t;*

*char s;*

*vector<double> temp;*

*vector<char> name;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*while (cin>>t>>s)*

*{ try*

*{*

*temp.push\_back(convert(t,s,'C'));*

*temp.push\_back(convert(t,s,'F'));*

*temp.push\_back(convert(t,s,'K'));*

*}*

*catch (const invalid\_argument& e)*

*{*

*cerr << e.what() << "\n";*

*}*

*catch (const logic\_error& le)*

*{*

*cerr << le.what() << "\n";*

*}*

*catch (...)*

*{*

*cerr << "Unknown error! Try again! \n";*

*}*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*}*

*cout <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < temp.size(); ++i)*

*{*

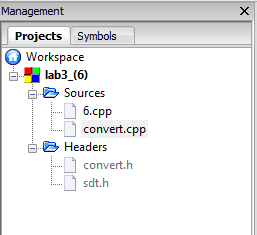
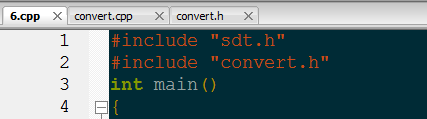
*printf("%4.2f\t", temp[i]);*

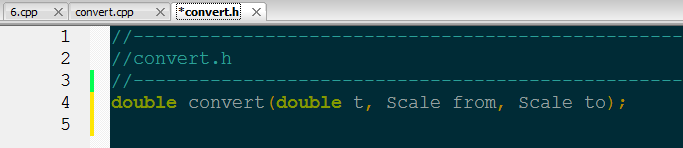
*if ((i+1)%3==0) cout <<"\n";*

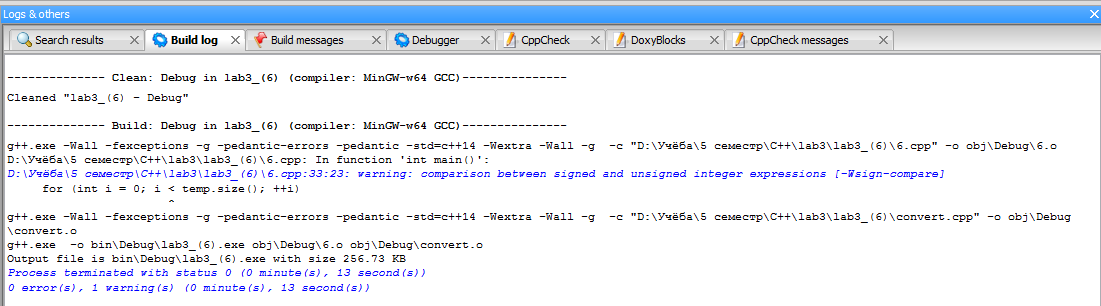
*}*

*}*

*6.* Создать (изменить) проект так, чтобы в нем были: основной файл исходного кода с функцией main(), заголовочный файл с объявлением функции convert(), файл реализации с определением функции convert().

**

**

**

***-------------- Clean: Debug in lab3\_(6) (compiler: MinGW-w64 GCC)---------------***

*Cleaned "lab3\_(6) - Debug"*

***-------------- Build: Debug in lab3\_(6) (compiler: MinGW-w64 GCC)---------------***

*g++.exe -Wall -fexceptions -g -pedantic-errors -pedantic -std=c++14 -Wextra -Wall -g -c "D:\Учёба\5 семестр\С++\lab3\lab3\_(6)\6.cpp" -o obj\Debug\6.o*

*D:\Учёба\5 семестр\С++\lab3\lab3\_(6)\6.cpp: In function 'int main()':*

*D:\Учёба\5 семестр\С++\lab3\lab3\_(6)\6.cpp:33:23: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]*

*for (int i = 0; i < temp.size(); ++i)*

*^*

*g++.exe -Wall -fexceptions -g -pedantic-errors -pedantic -std=c++14 -Wextra -Wall -g -c "D:\Учёба\5 семестр\С++\lab3\lab3\_(6)\convert.cpp" -o obj\Debug\convert.o*

*g++.exe -o bin\Debug\lab3\_(6).exe obj\Debug\6.o obj\Debug\convert.o*

*Output file is bin\Debug\lab3\_(6).exe with size 256.73 KB*

*Process terminated with status 0 (0 minute(s), 13 second(s))*

*0 error(s), 1 warning(s) (0 minute(s), 13 second(s))*

7. Добавить в заголовочный файл определение перечисления Scale, заменить тип параметров to и from функции convert() на Scale.

Содержание заголовочного файла:

*//------------------------------------------------------------------------------*

*//convert.h*

*//------------------------------------------------------------------------------*

*enum Scale*

*{*

*Kelvin,*

*Celsius,*

*Fahrenheit*

*};*

*double convert(double t, Scale from, Scale to);*

*char from\_Scale(Scale s); //функция, конвертирующая введённое значение в шкалу*

*//Scale*

Тогда функция convert() примет вид:

*double convert(double t, Scale from, Scale to)*

*{ double K;*

*if (to==from) return t;*

*switch (from)*

*{*

*case Kelvin: K=t; break;*

*case Celsius: K=t+273.15; break;*

*case Fahrenheit: K=(5\*(t-32))/9+273.15; break;*

*//default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*if (K<0) throw logic\_error("Too low temperature!\n");*

*if (K>=0)*

*{ switch (to)*

*{*

*case Celsius: return (K-273.15); break;*

*case Fahrenheit: return ((9\*(K-273.15))/5+32); break;*

*case Kelvin: return K; break;*

*}*

*}*

*throw exception();*

*}*

Функция конвертации из char в Scale:

*Scale to\_scale(char s)*

*{ switch (s)*

*{case 'c':*

*case 'C': return Celsius; break;*

*case 'F':*

*case 'f': return Fahrenheit; break;*

*case 'K':*

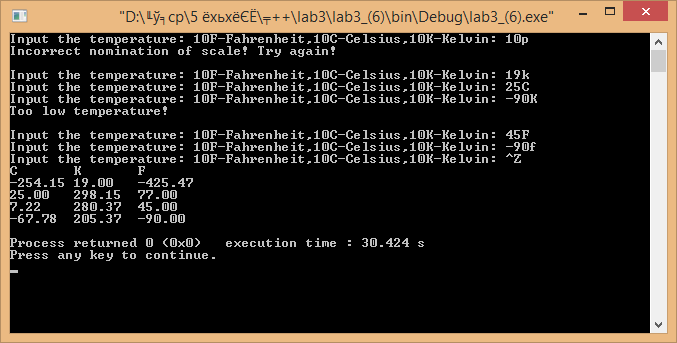
*case 'k': return Kelvin; break;*

*default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*}*

Результат работы:



8. Возбуждение исключения порождается invalid\_argument(см. пункт 5). Этим решается проблема неопределенного поведения функции при некорректном переводе из одной шкалы в другую. Исключение должно информировать программиста, заставляя его добавить в код недостающие для перевода шкалы.

9. Добавить в заголовочный файл определение структуры Temperature, представляющей температуру со шкалой, с конструктором Temperature(double value, char scale). Реализацию конструктора поместить в отдельный файл исходного кода в проекте.

В заголовочный файл добавлено описание структуры:

*struct Temperature //Описание структуры*

*{ Temperature(double value,char scale);*

*Temperature();*

*double value;*

*char scale;*

*};*

и в файл ctructure.cpp реализация конструктора:

*Temperature::Temperature(double value, char scale)*

*{*

*this->scale = scale;*

*this->value = value;*

*}*

10. Модернизировать основную программу: использовать Temperature вместо отдельных переменных для значения и шкалы, а также для хранения в векторе.

*int main()*

*{ vector<Temperature> temp;*

*Temperature input;*

*input=Temperature();*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*while (cin >>input.value >>input.scale)*

*{ if (!cin) break;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*try*

*{*

*temp.push\_back(convert(input,'C'));*

*temp.push\_back(convert(input,'K'));*

*temp.push\_back(convert(input,'F'));*

*}*

*catch (const invalid\_argument& e)*

*{*

*cerr << e.what() << "\n";*

*}*

*catch (const logic\_error& le)*

*{*

*cerr << le.what() << "\n";*

*}*

*catch (const invalid\_argument& le)*

*{*

*cerr << le.what() << "\n";*

*}*

*catch (...)*

*{*

*cerr << "Unknown error! Try again! \n";*

*}*

*}*

*cout <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < (temp.size()); ++i)*

*{*

*printf("%4.2f\t", temp[i]);*

*if ((i+1)%3==0) cout <<"\n";*

*}*

*}*

11. Перегрузить операторы ввода и вывода в поток для перечисления Scale и для структуры Temperature (итого четыре новые функции). Задействовать их в основной программе.

1) Ввод и вывод в поток структуры Temperature

В заголовочный файл convert.h добавляем:

*istream& operator>>(istream& in, Temperature& t);//объявление ф-ции польз.ввода*

*ostream& operator<<(ostream& out, const Temperature& t); //объявление ф-ции польз. вывода*

В файл ctructure.cpp добавляем реализацию:

*istream& operator>>(istream& in, Temperature& t)*

*{*

*in >> t.value >> t.scale;*

*return in;*

*}*

*ostream& operator<<(ostream& out, const Temperature& t)*

*{*

*out << t.value << t.scale;*

*return out;*

*}*

2) Ввод и вывод в поток перечисления Scale

В заголовочный файл convert.h добавляем:

*istream& operator>>(istream& in, Scale& s);*

*ostream& operator<<(ostream& out, const Scale& s);*

В файл ctructure.cpp добавляем реализацию:

*istream& operator>>(istream& in, Scale& s)*

*{ char value;*

*in >> value;*

*switch (value)*

*{*

*case 'K':*

*case 'k': s = Kelvin; break;*

*case 'C':*

*case 'c': s = Celsius; break;*

*case 'F':*

*case 'f': s = Fahrenheit; break;*

*default: in.setstate(:ios\_base::failbit);//сообщаем об ошибке*

*}*

*return in;*

*}*

*ostream& operator<<(ostream& out, const Scale& s)*

*{ switch (s)*

*{*

*case Fahrenheit: out << 'F'; break;*

*case Celsius: out << 'C'; break;*

*case Kelvin: out << 'K'; break;*

*}*

*return out;*

*}*

3) Задействовать их в основной программе

Делаем изменения в заголовочном файле и соответствующие в файле реализации, чтобы использовать Scale вместо char:

*struct Temperature //Описание структуры*

*{ Temperature(double value,* ***Scale*** *scale);*

*Temperature();*

*double value;*

***Scale*** *scale;*

*};*

*Temperature convert(Temperature t,* ***Scale*** *to); //объявление функции конвертации*

Функция конвертации будет реализована так:

*Temperature convert(Temperature t, Scale to)*

*{ double K;*

*Scale from=t.scale;*

*Temperature result;*

*if (to==from)*

*{result.value=t.value;*

*result.scale=to;*

*}*

*switch (from)*

*{*

*case Kelvin: K=t.value; break;*

*case Celsius: K=t.value+273.15; break;*

*case Fahrenheit: K=(5\*(t.value-32))/9+273.15; break;*

*default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*if (K<0) throw logic\_error("Too low temperature!\n");*

*if (K>=0)*

*{ switch (to)*

*{*

*case Celsius:*

*result.value=(K-273.15);*

*result.scale=to;*

*break;*

*case Fahrenheit:*

*result.value=(9\*(K-273.15))/5+32;*

*result.scale=to;*

*break;*

*case Kelvin:*

*result.value=K;*

*result.scale=to;*

*break;*

*default: throw invalid\_argument("Incorrect nomination of scale! Try again!\n");*

*}*

*}*

*return result;*

*throw exception();*

*}*

И, наконец, основная программа:

*int main()*

*{ vector<Temperature> temp;*

*Temperature input;*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*while (cin >>input)*

*{ if (!cin) break;*

*try*

*{*

*temp.push\_back(convert(input,Celsius));*

*temp.push\_back(convert(input,Kelvin));*

*temp.push\_back(convert(input,Fahrenheit));*

*}*

*catch (const invalid\_argument& e)*

*{*

*cerr << e.what() << "\n";*

*}*

*catch (const logic\_error& le)*

*{*

*cerr << le.what() << "\n";*

*}*

*catch (const invalid\_argument& le)*

*{*

*cerr << le.what() << "\n";*

*}*

*catch (...)*

*{*

*cerr << "Unknown error! Try again! \n";*

*}*

*cout << "Input the temperature: 10F-Fahrenheit,10C-Celsius,10K-Kelvin: ";*

*}*

*cout <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < (temp.size()); ++i)*

*{*

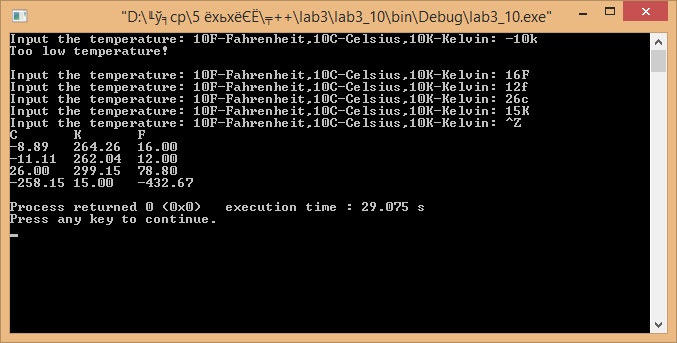
*printf("%4.2f\t", temp[i]);*

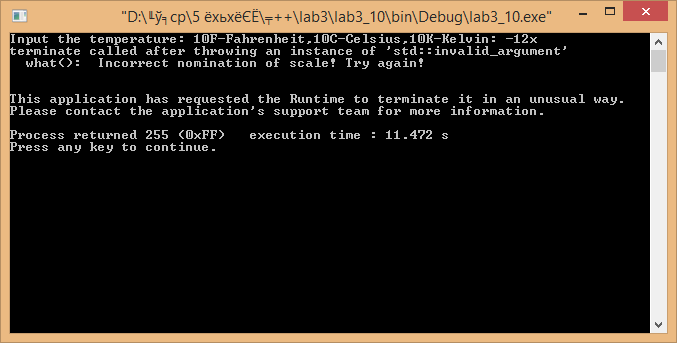
*if ((i+1)%3==0) cout <<"\n";*

*}*

*}*

Результат работы программы:





12. Перегрузить оператор сравнения для двух температур (<), сложения двух температур (+), деления температуры на число (/) и соответствующие им операторы сокращенного присваивания (+=, /=). Задействовать их для того, чтобы по окончании ввода данных печатать наибольшую, наименьшую и среднюю температуру. 1) Сравнение

*bool Temperature::operator<(const Temperature &scale1) const*

*{*

*if (this->scale == scale1.scale)*

*{*

*return this->value < scale1.value;*

*}*

*else*

*{*

*Temperature temp = convert(scale1, this->scale);*

*return this->value < temp.value;*

*}*

2) Сокращённое присваивание

*Temperature& Temperature::operator+=(const Temperature& scale1)*

*{*

*if (this->scale == scale1.scale)*

*{*

*this->value += scale1.value;*

*}*

*else*

*{*

*Temperature temp = convert(scale1, this->scale);*

*this->value += temp.value;*

*}*

*return \*this;*

*}*

*Temperature& Temperature::operator/=(int d)*

*{*

*this->value /= d;*

*return \*this;*

*}*

3) Сложение и деление

Просто выразить через операторы += и /=:

*const Temperature& Temperature::operator+(const Temperature& scale1) const*

*{*

*return (Temperature)(\*this) += scale1;*

*}*

*const Temperature& Temperature::operator/(int d) const*

*{*

*return (Temperature)(\*this) /= d;*

13. Выделить вычисление наибольшей, наименьшей и средней температур в функцию. Типы и способы передачи параметров выбрать самостоятельно.

*Temperature max\_temp(vector<Temperature> temp)*

*{*

*Temperature max\_t = Temperature(0, Scale::Kelvin);*

*for (int i = 0; i < temp.size(); i++)*

*{*

*if (temp[i] > max\_t)*

*{*

*max\_t = temp[i];*

*}*

*}*

*return max\_t;*

*}*

*//------------------------------------------------------------------------------*

*Temperature mid\_temp(vector<Temperature> temp)*

*{ Temperature mean = Temperature(0, Scale::Kelvin);*

*for (int i = 0; i < temp.size(); i++)*

*{*

*mean += temp[i];*

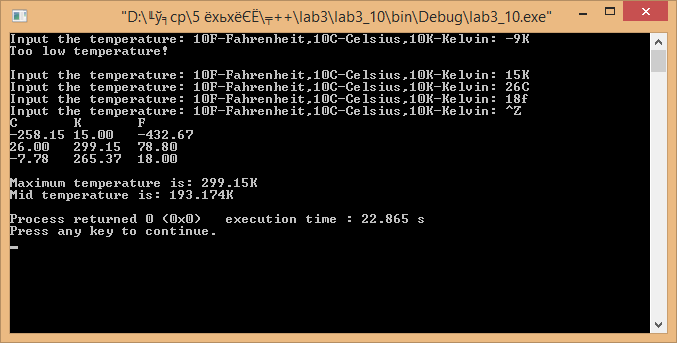
*}*

*mean /= temp.size();*

*return mean;*

*}*

Пример работы программы:



14. Добавить сохранение всех введенных температур в текстовый файл перед окончанием работы программы.

*ofstream out("convert\_temp.txt", ios\_base::trunc); //перед записью файл очищается*

*out <<"C \t" <<"K \t" <<"F \t \n";*

*for (int i = 0; i < (temp.size()); ++i)*

*{*

*out <<temp[i]<<"\t"*

*if ((i+1)%3==0) out <<"\n";*

*}*

*out <<"\nMaximum temperature is: " << max\_temp(temp\_K) <<"\n";*

*out << "Mid temperature is: " << mid\_temp(temp\_K)<<"\n";*

*out.close();*

Результат работы программы:

