1.

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

main()
{
	char degree = 248;
	float firstTemp = -5.4, secondTemp = 124.67, thirdTemp = 305.15; // Celius,
Farenheight, Kelvin
	float firstFarenheightConversion, firstKelvinConversion,
firstRankineConversion;

	firstFarenheightConversion = ((1.8) * firstTemp) + 32;
	firstKelvinConversion = (firstTemp) + 273.15;
	firstRankineConversion = firstFarenheightConversion + 459.67;

	cout << setprecision(2) << fixed << "First Constant Value = " << firstTemp
<< degree << "C\n"
		<< degree << "F = " << firstFarenheightConversion << degree << "F\n"
		<< degree << "K = " << firstKelvinConversion << degree << "K\n"
		<< degree << "R = " << firstRankineConversion << degree << "R\n";

	float secondFarenheightConversion, secondKelvinConversion,
secondRankineConversion;

	secondFarenheightConversion = 5 * (secondTemp - 32) / 9;
	secondKelvinConversion = secondFarenheightConversion + 273.15;
	secondRankineConversion = secondTemp + 459.67;

	cout << "\nSecond Constant Value = " << secondTemp << degree << "F\n"
		<< degree << "F = " << secondFarenheightConversion << degree << "C\n"
		<< degree << "K = " << secondKelvinConversion << degree << "K\n"
		<< degree << "R = " << secondRankineConversion << degree << "R\n";

	float thirdFarenheightConversion, thirdCelciusConversion,
thirdKelvinConversion, thirdRankineConversion;

	thirdFarenheightConversion = (thirdTemp - 491.67) * 5 / 9;
	thirdCelciusConversion = thirdFarenheightConversion;
	thirdKelvinConversion = ((1.8) * thirdFarenheightConversion) + 32;
	thirdRankineConversion = thirdTemp * 0.5555555556;
```

```
    cout << "\nThird Constant Value = " << thirdTemp << degree << "R\n"
         << degree << "C = " << thirdFarenheightConversion << degree << "C\n"
         << degree << "F = " << thirdKelvinConversion << degree << "F\n"
         << degree << "R = " << thirdRankineConversion << degree << "R\n";

    getch();
}
```

Output:

```
First Constant Value = -5.40°C
°F = 22.28°F
°K = 267.75°K
°R = 481.95°R

Second Constant Value = 124.67°F
°F = 51.48°C
°K = 324.63°K
°R = 584.34°R

Third Constant Value = 305.15°R
°C = -103.62°C
°F = -154.52°F
°R = 169.53°R
```

In this program, we are tasked to convert temperature values to different units of temperature. To do so, we are given the equations to certain conversions such as:

$$°F = \frac{9}{5}°C + 32$$

$$°K = °C + 273.15$$

$$°R = °F + 459.67$$

However, we cannot use these equations for all the conversions that we are going to do. In this case, we are going to derive these 3 equations to find the other units.

$$°C = \frac{5(°F - 32)}{9}$$

$$°K = \frac{5(°F - 32)}{9} + 273.15$$

$$°C = °K - 273.15$$

$$°F = \frac{9(°K - 273.15)}{5} + 32$$

$$°C = \frac{5(°R - 491.67)}{9}$$

$$°F = °R - 459.67$$

$$°K = \frac{5°R}{9}$$

We will use these derived equations in finding the missing conversions in the supplementary problem.

One more feature that I used in the code is using <iomanip>. I used iomanip to manipulate the output of the numbers in the terminal. In this case, I used setprecision(2) << fixed where it will only show 2 decimals, and adding fixed to it will only fix it to 2 decimals no matter how many decimals are there.

2.

```
#include <iostream>
#include <conio.h>
#include <cmath>

using namespace std;

main()
{
    cout << "N\t2^n\n"
         << "0\t" << pow(2, 0) << "\n"
         << "1\t" << pow(2, 1) << "\n"
         << "2\t" << pow(2, 2) << "\n"
         << "3\t" << pow(2, 3) << "\n"
         << "4\t" << pow(2, 4) << "\n"
         << "5\t" << pow(2, 5) << "\n"
         << "6\t" << pow(2, 6) << "\n"
         << "7\t" << pow(2, 7) << "\n"
         << "8\t" << pow(2, 8) << "\n";
    getch();
}
```

Output:

```
N         2^n
0         1
1         2
2         4
3         8
4         16
5         32
6         64
7         128
8         256
```

In the second supplementary problem, I used the library cmath to utilize the pow function. The pow function is used to indicate exponents in numbers. The format is as follows: pow(base, exponent), where the base is the base number and the exponent is the value raised to the base number. For example, we can write 2^4 as pow(2, 4) as we write it in C++ programs.

```
#include <iostream>
#include <conio.h>
#include <cmath>

using namespace std;

main()
{
    float a, T, m, g; // a = initial acceleration T - thrust in Newton m - mass 
in kg g - acceleration caused by gravity in m/s^2

    T = 6e6;
    g = 9.81;
    m = 5e4;

    a = (T - (m * g)) / m;

    cout << "The initial acceleration, a, of a rocket fired from earth , with an 
initial thrust, T, is given by this formula:\n a = (T - mg) / m\n where:\n a = 
initial acceleration \n T = thrust in Newton\n m = mass in kg\n g = acceleration 
caused by gravity in m/s^2"
         << "\nWrite and run a program that will determine the initial 
acceleration of a rocket having a mass of 5X10^4 kg and an initial thrust of 
6X10^5 Newtons. The Value of g is 9.81m/s^2.\n\n";
    cout << "The initial acceleration is " << a << "m/s^2.";
    getch();
```

```
}
```

```
The initial acceleration, a, of a rocket fired from earth , with an initial thrust, T, is given by this formula:
 a = (T - mg) / m
 where:
 a = initial acceleration
 T = thrust in Newton
 m = mass in kg
 g = acceleration caused by gravity in m/s^2
Write and run program that will determine the initial acceleration of a rocket having a mass of 5X10^4 kg and an initial thrust of 6X10^5
 Newtons. The Value of g is 9.81m/s^2.

The initial acceleration is 110.19m/s^2.
```

In the last supplementary problem, we used scientific notations in indicating our values. We write scientific notations as aeb, where a is the base number, e is denoted by $\times 10$, and b is the exponent. For example, we can write 6×10^4 as *6e4.* The formula provided is used in finding the initial acceleration.