**mgr Agata Emilia WITEK**

University of Szczecin: Faculty of Humanities, Institute of Philology *(graduate)* and
Faculty of Mathematics and Physics, Department of Informatics & Technical Education *(graduate)*

Uniwersytet Szczeciński: Wydział Nauk Humanistycznych, Instytut Filologii *(absolwentka)* oraz
Wydział Matematyczno-Fizyczny, Katedra Edukacji informatycznej i Technicznej *(absolwentka)*

# CONVERSION WITHIN A DECIMAL AND BINARY NUMERAL SYSTEMS IN C$^{++}$ LANGUAGE

## Abstract

**Introduction and aim:** Converting numbers from one number system to another is an important skill, used commonly in millions of computers all over the world. However, even a beginner programmer should face the problem of converting numbers with the support of the programming language *C*$^{++}$. This article shall briefly described two numeral systems, and after a short programming introduction in *C*$^{++}$ the source code would be offered which easily converts a numbers within both systems.
**Material and methods**: After a short introduction of programming in *C*$^{++}$, there was proposed the program source code, which easily converts a numbers within both systems. To create the program the user will need some basic knowledge of the syntax of *C*$^{++}$, a wide range of books and courses available in the market.
**Results**: It is presented the program is written and compiled in *Orwell Dev-C*$^{++}$ *5.1.1.0.*
**Conclusion:** Conversion of numbers within the two most common numerical systems is widespread, so the ability to create the source code, for example, in *C*$^{++}$.

**Keywords:** Decimal and the binary numeral system, programming, *C*$^{++}$ language.

(Received: 11.12.2012; Revised: 20.08.2013; Accepted: 31.08.2013)

# *KONWERSJA W JĘZYKU C$^{++}$ MIĘDZY SYSTEMAMI DZIESIĄTKOWYM I BINARNYM*

## *Streszczenie*

***Wstęp i cel:*** *Konwersja liczb w obrębie dwóch najbardziej rozpowszechnionych systemów liczbowych jest stosowana codziennie w milionach komputerowych umysłów na całym świecie. Jednak i początkujący programista może zmierzyć się z problemem konwersji liczb, przy wsparciu języka programowania C*$^{++}$. *W niniejszym artykule pokrótce zostaną opisane dwa wspomniane systemy liczbowe, zaś po krótkim wstępie dotyczącym programowania w języku C*$^{++}$, *zaproponowany oraz pokrótce opisany zostanie kod źródłowy programu, który w prosty sposób konwertuje liczby w obrębie obu systemów.*
***Materiał i metody:*** *Po krótkim wprowadzeniu do programowania w C*$^{++}$, *został zaproponowany kod źródłowy programu, który łatwo przekształca liczby ciągu obu systemów. Aby utworzyć program, użytkownik musi posiadać podstawową wiedzę o składni C*$^{++}$, *szeroki wybór podręczników i kursów dostępnych na rynku.*
***Wyniki:*** *Przedstawiono program napisany i skompilowany w Orwell Dev-C*$^{++}$ *5.1.1.0.*
***Wniosek:*** *Konwersja liczb w dwóch najbardziej powszechnych systemach numerycznych jest powszechne znana, to zdolność do tworzenia kodu źródłowego na przykład w C*$^{++}$.

***Słowa kluczowe:*** *Decymalny i binarny system numeryczny, programowanie, język C*$^{++}$.

*(Otrzymano: 11.12.2012; Zrecenzowano: 20.08.2013; Zaakceptowano: 31.08.2013)*

*© A.E. Witek 2013*

## 1. Decimal and binary numeral systems

### 1.1 Numeral system in mathematical expression

The decimal numeral system (also known as *base ten* or *denary*) has ten as its base. It is most commonly used in modern world. The exemplary number 21 is given below as decimal:

$$21_{(X)} = 2 \cdot 10^1 + 1 \cdot 10^0 = 20 + 1 = 21_{(X)}. \tag{1}$$

The binary numeral system, (also called base-2 number system) is only using two symbols: 0 and 1. More specifically, the usual base-2 system is a positional notation with a radix of 2. The binary system is used internally by almost all modern computers.

### 1.2 Conversion within numeral systems

To convert an decimal integer (or an integer part of a number) to a binary number, it has to be divided repeatedly by 2, until down to 1, leaving a remainder (1) or lack of the remainder (0). Then the remainders should be read from bottom to top. The exemplary number 31 can be converted as following:

$$\begin{array}{l} 30 : 2 \mid 0 \\ 15 : 2 \mid 1 \\ 7 : 2 \ \ \mid 1 \\ 3 : 2 \ \ \mid 1 \\ 1 : 2 \ \ \mid 1 \end{array} \tag{2}$$

The binary equivalent for 30 is 11110.

To convert a binary number to a decimal for the integer part, the appropriate powers of 2 has to be added beginning from the smallest part of the number, as in the exemplary following:

$$10101_{(II)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 0 + 4 + 0 + 1 = 21_{(X)}. \tag{3}$$

## 2. The short introduction to C$^{++}$ language

### 2.1 The C$^{++}$ language

The *C*$^{++}$ is a continuation and an extension of the C programming language. It was designed by Bjarne Stroustrup and allows both structured and object-oriented programming. Each program is opened by a reference to the library (by the command *# include*). This is a library brings a series of operations, for example, it enables the input/output operations.

Each program also encloses the main function (*main()*), that contents the body of the program, which are commands executed and performed by the program. The instructions executed in the main function, as well as in other, smaller features are framed by curly braces (*{}*). Like any other language, the C$^{++}$ has its own syntax. Ending each statement with a semicolon is also one of its elements.

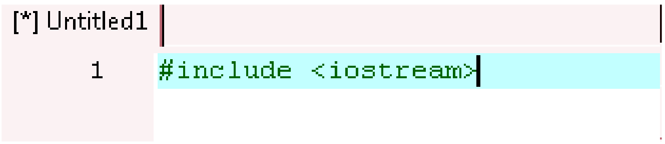
Basic commands, control statements, types of constants, variables used by the programmer, operators, or the way to create functions, have been described
in many widely available manuals and books. Basic information contained in them allowed me to develop this program.

To write a program that will convert the numbers, the user should become familiar with different types of variables and instructions, also operators and functions shall be used. There are several books that introduce the basics of $C^{++}$, so in this article, these instructions will be omitted.

The program is written and compiled in *Orwell Dev-C++ 5.1.1.0* on *GNU*.

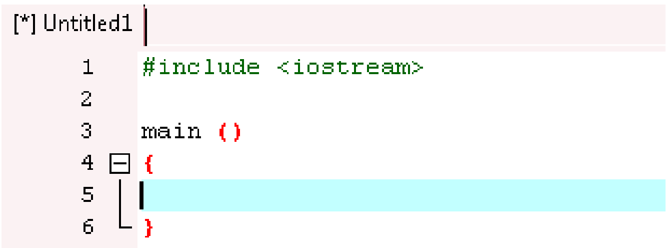
## 2.2 The program source code

To create a program, first enter a specific library. During compilation, the transformation of the program language to one that is readable by machine, it will also load the library (it is also called linking) (Fig. 1 and Fig. 2).



```
#include <iostream>
```

Fig. 1. Linking <*iostream*> library
*Source: Made by the Author*



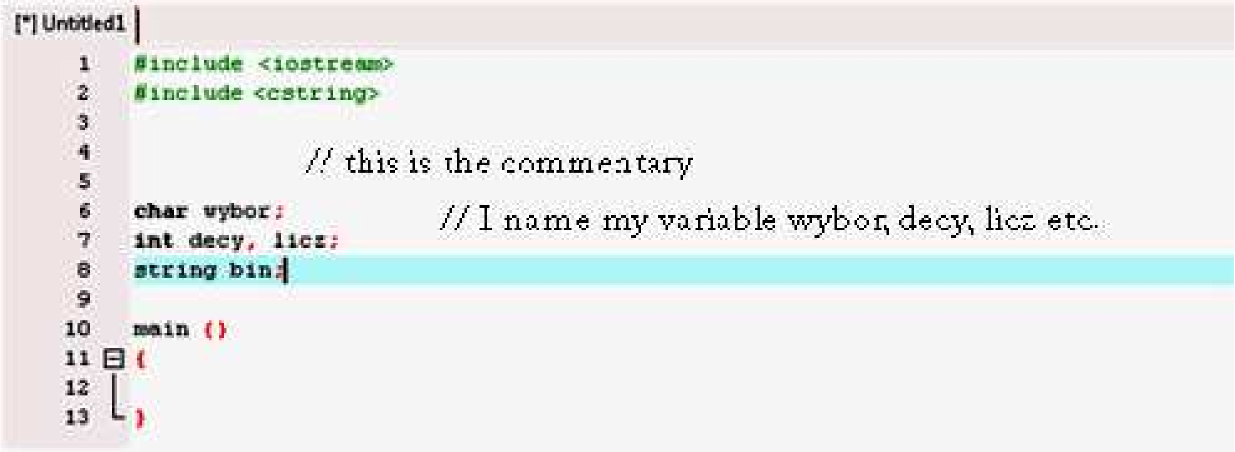
```
#include <iostream>

main ()
{

}
```

Fig. 2. The main function. In between the braces is entered everything what the program is to perform
*Source: Made by the Author*

The program will use variables of different types. We will need integers, which will be introduced by a variable *int*. If the letter is needed, a variable *char* should be declared.

Declaration of variables (we declare that we will use them) is defined at the same moment (we define their type - integer, or floating point, strings, etc..).

In this program, the declaration (and the definition) will take place before the function *main()*, but the language features allow this initiation anywhere in the program (Fig. 3):

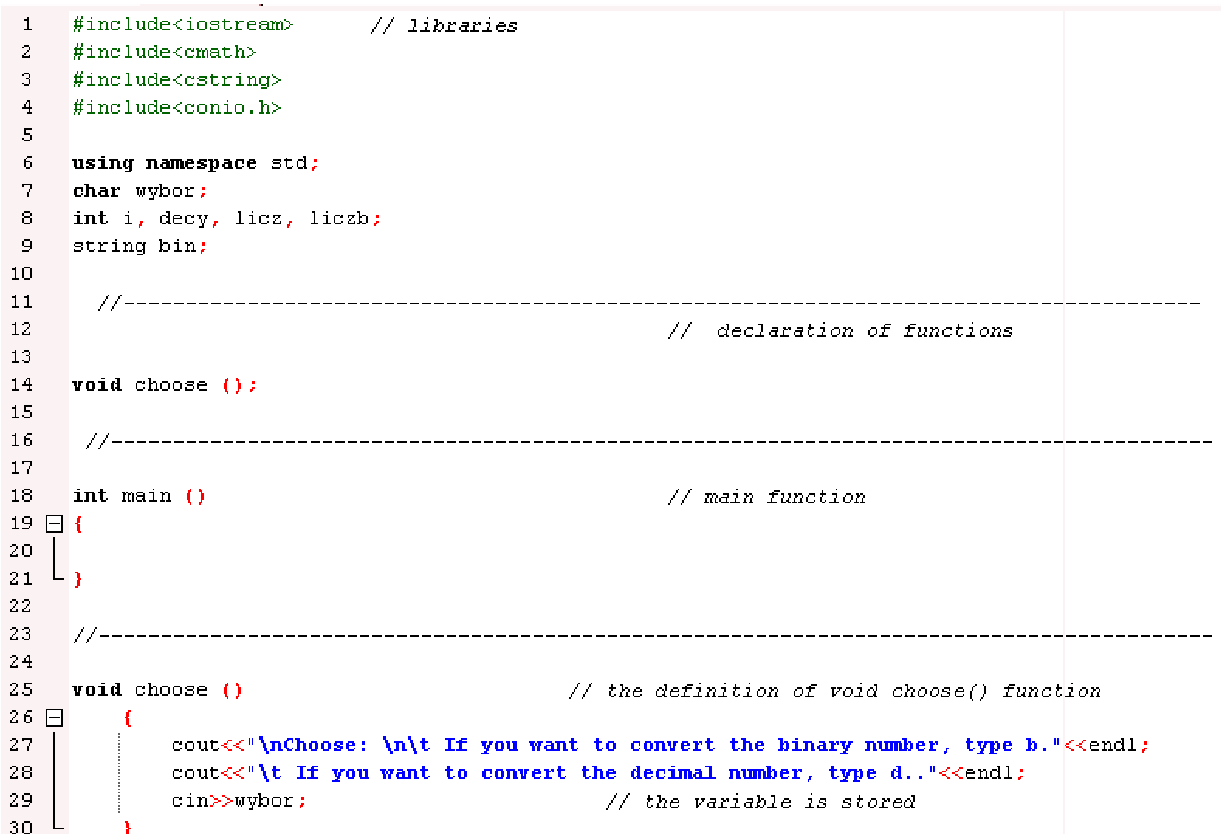


```
#include <iostream>
#include <cstring>

                // this is the commentary

char wybor;             // I name my variable wybor, decy, licz etc.
int decy, licz;
string bin;

main ()
{

}
```

Fig. 3. The variables initiation
*Source: Made by the Author*

## 2.3 An exemplary conversion program in $C^{++}$

In the program the user will have the choice to convert numbers from binary to decimal numeral systems and inversely. To allow such a choice, the function is written. Function declaration is placed before the function *main ()* as shown in the figure (Fig. 4), and its definitions can be placed behind the body of *main ()* function:



```
#include<iostream>      // libraries
#include<cmath>
#include<cstring>
#include<conio.h>

using namespace std;
char wybor;
int i, decy, licz, liczb;
string bin;

  //-------------------------------------------------------------------------------------------
                                              //  declaration of functions

void choose ();

 //---------------------------------------------------------------------------------------------

int main ()                                   // main function
{

}

//---------------------------------------------------------------------------------------------

void choose ()                        // the definition of void choose() function
    {
        cout<<"\nChoose: \n\t If you want to convert the binary number, type b."<<endl;
        cout<<"\t If you want to convert the decimal number, type d.."<<endl;
        cin>>wybor;                           // the variable is stored
    }
```

Fig. 4. Placing the declaration and definition of function. To use freely with *cin* and *cout* commands, type the command *using namespace std;* which allows you to use a standard namespace
*Source: Made by the Author*

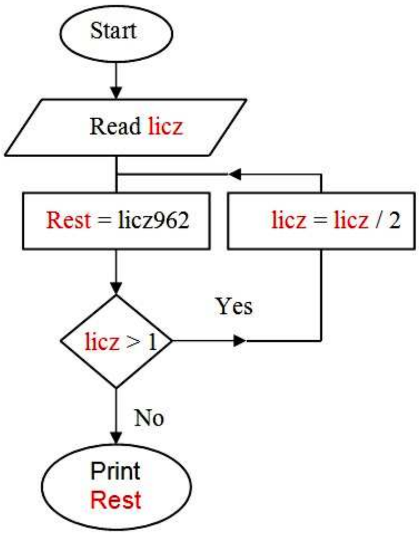
## 2.4 Definitions of functions

There are two definitions of converting functions. The first of these will convert a number from decimal to binary, the other inversely. Conversion from decimal to binary is presented in the form of block diagram (Fig. 5), and convert numbers from decimal to binary system, is presented in the figure 6.

In the function *binarna* I define the rest, as the result of modulo two, and enter a loop *if* that from the moment when the count value is greater than one, the program performs the division by two and using the recursion calls the binary again. When the count value is less than or equal to one, then the loop is interrupted and a series of numbers is displayed (the last saved character will be displayed first).

Note that the function is introduced recursive function. This is a function, which in its own definition (body) calls itself.

In the function decimal an variable named *power* is declared. The highest power is one less than the number in the binary system, which converts the number in decimal. There was also declared integer *tab*, which is an array of as many characters, as in the variable *power* (*int tab[power]*).



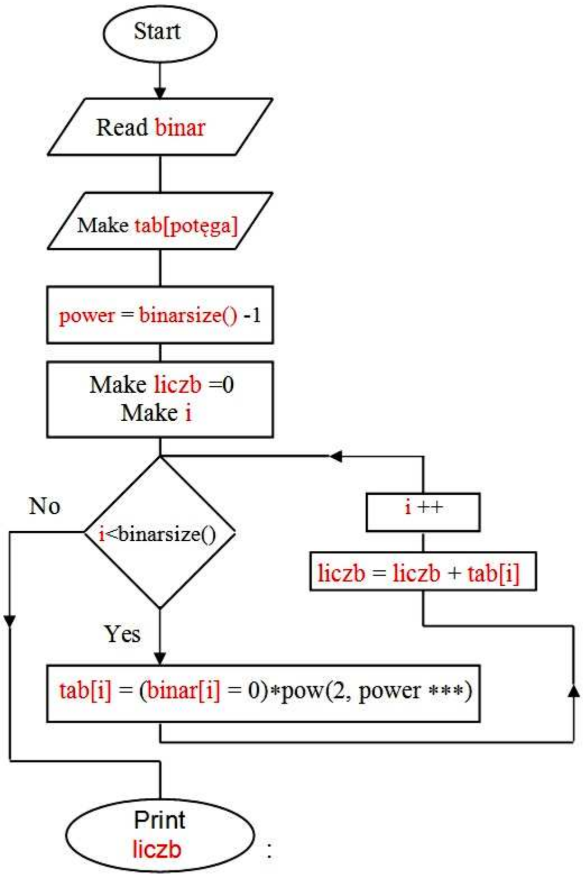


Fig. 5. The block diagram of function *binarna*
*Source: Made by the Author*

Fig. 6. The block diagram of function *decimal*
*Source: Made by the Author*

To convert a number of forms in a *for* loop for each item number, starting from zero *(int i = 0)*, until there will be one number less than the declared size of the binary number, then follow the commands and step i+1. For each character array multiply the mark and the power of the two, the result is the sum of the numbers and as a result of the multiplication.

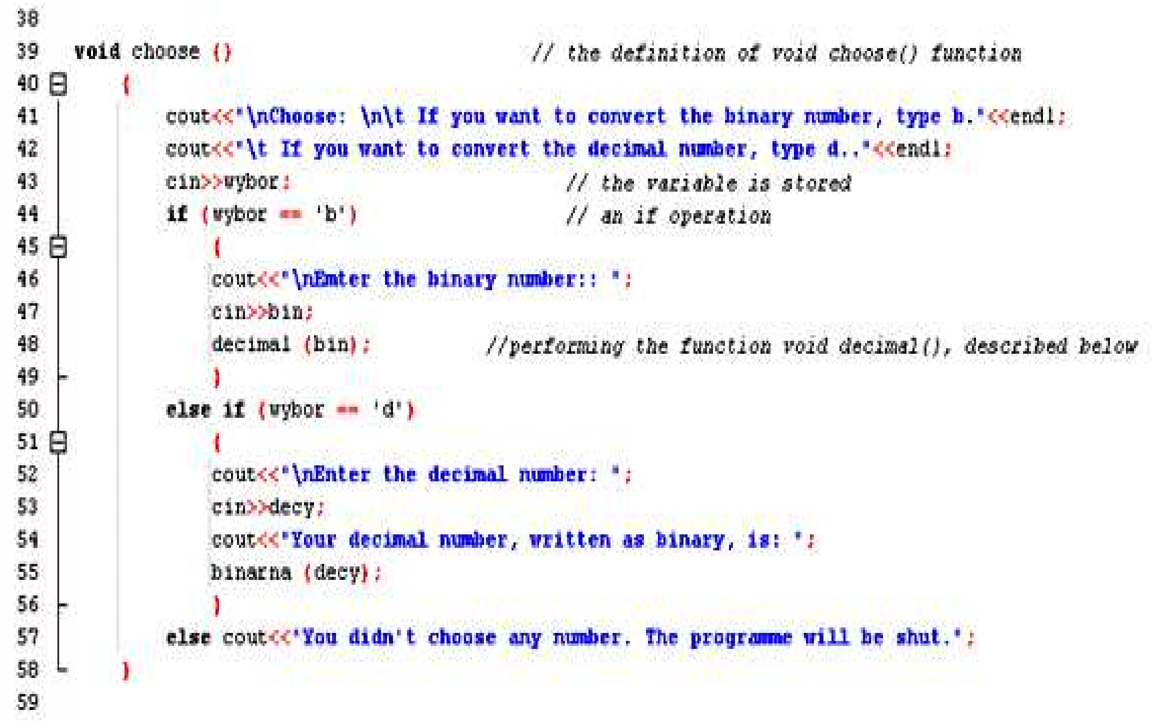
When condition are finished, the loop will be broken and the results would be viewed, using cout.

### 2.5 Content of main ()

In the body of the function *main()* function, there will be called a *choose()*, which places the appropriate function calls. Content of the function is shown in the figure 7.

In the function, the program asks the user to select the method of converting. The user has to write the letter. Somebody can write *b* for the binary system, *d* for decimal system, may also write the rest of the characters from the keyboard. Therefore, if the letter *b* is typed (so if the condition is made for the first loop), the program stores a binary number. It should then perform the function of switching on this variable. At this point the program goes back to the definition of the function to repeat the action, but the new variable substitutes a variable bin. If the condition of the first loop is not fulfilled, the program goes on and encounters an alternate loop, in which a decimal number is conversed.

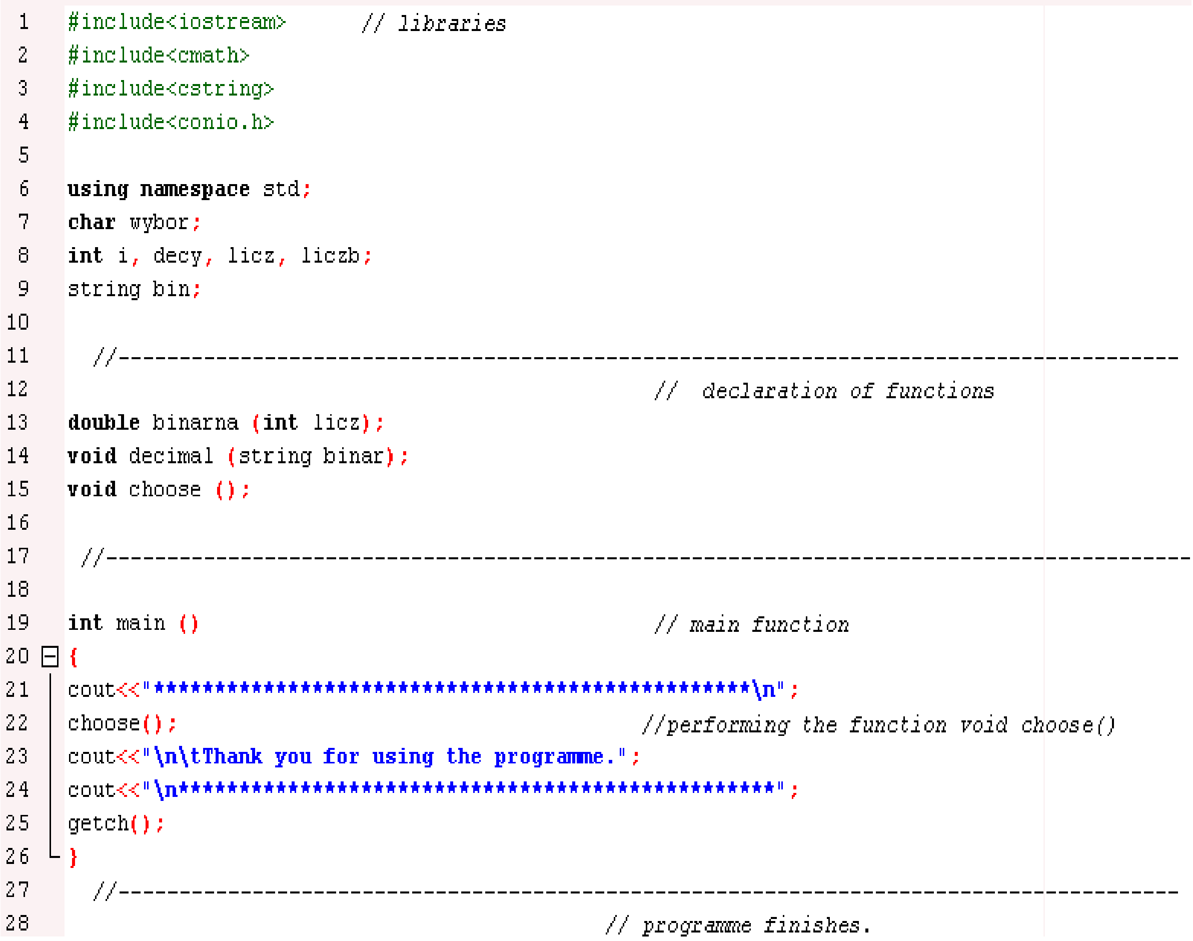
When no condition is not fulfilled, then the program will perform alternative *default* statement, which is designed to display a message that user has not selected any number and to shut the program down.



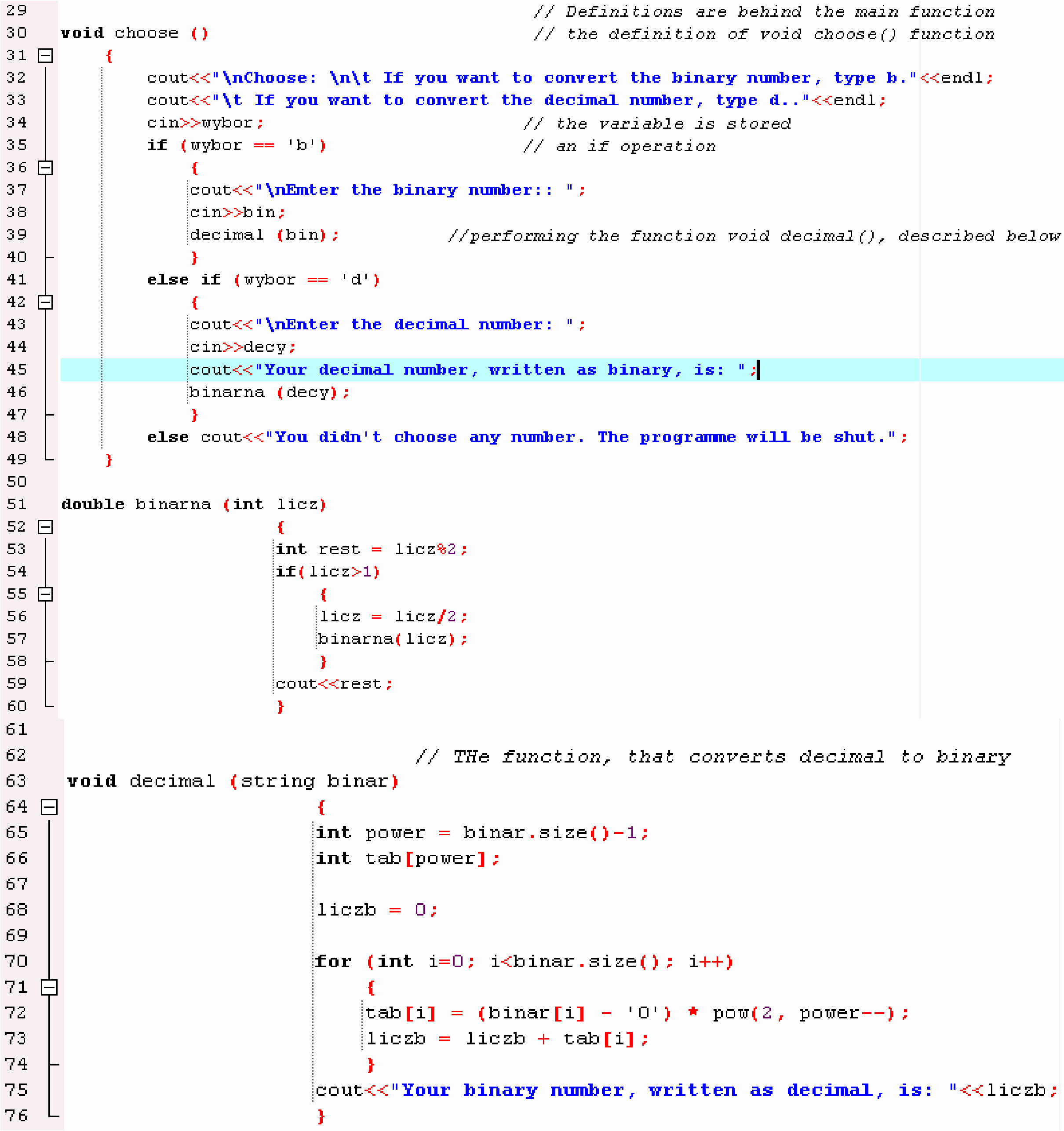
```

void choose ()                                    // the definition of void choose() function
    {
       cout<<"\nChoose: \n\t If you want to convert the binary number, type b."<<endl;
       cout<<"\t If you want to convert the decimal number, type d.."<<endl;
       cin>>wybor;                                // the variable is stored
       if (wybor == 'b')                          // an if operation
           {
           cout<<"\nEmter the binary number:: ";
           cin>>bin;
           decimal (bin);                //performing the function void decimal(), described below
           }
       else if (wybor == 'd')
           {
           cout<<"\nEnter the decimal number: ";
           cin>>decy;
           cout<<"Your decimal number, written as binary, is: ";
           binarna (decy);
           }
       else cout<<"You didn't choose any number. The programme will be shut.";
    }

```

Fig. 7. The body of the  function void choose ()
*Source: Made by the Author*

Throughout the program source code is shown on the figure 8 below:



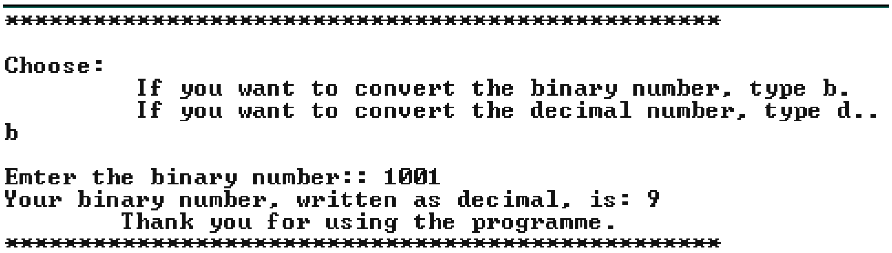
```
#include<iostream>      // libraries
#include<cmath>
#include<cstring>
#include<conio.h>

using namespace std;
char wybor;
int i, decy, licz, liczb;
string bin;

  //---------------------------------------------------------------------------------------
                                              //  declaration of functions
double binarna (int licz);
void decimal (string binar);
void choose ();

 //-----------------------------------------------------------------------------------------

int main ()                                   // main function
{
cout<<"*************************************************\n";
choose();                                    //performing the function void choose()
cout<<"\n\tThank you for using the programme.";
cout<<"\n*************************************************";
getch();
}
  //---------------------------------------------------------------------------------------
                                          // programme finishes.
```



```
                                              // Definitions are behind the main function
void choose ()                                // the definition of void choose() function
    {
        cout<<"\nChoose: \n\t If you want to convert the binary number, type b."<<endl;
        cout<<"\t If you want to convert the decimal number, type d.."<<endl;
        cin>>wybor;                           // the variable is stored
        if (wybor == 'b')                     // an if operation
            {
            cout<<"\nEmter the binary number:: ";
            cin>>bin;
            decimal (bin);          //performing the function void decimal(), described below
            }
        else if (wybor == 'd')
            {
            cout<<"\nEnter the decimal number: ";
            cin>>decy;
            cout<<"Your decimal number, written as binary, is: ";
            binarna (decy);
            }
        else cout<<"You didn't choose any number. The programme will be shut.";
    }

double binarna (int licz)
                    {
                    int rest = licz%2;
                    if(licz>1)
                        {
                        licz = licz/2;
                        binarna(licz);
                        }
                    cout<<rest;
                    }

                                    // THe function, that converts decimal to binary
void decimal (string binar)
                      {
                      int power = binar.size()-1;
                      int tab[power];

                      liczb = 0;

                      for (int i=0; i<binar.size(); i++)
                          {
                          tab[i] = (binar[i] - '0') * pow(2, power--);
                          liczb = liczb + tab[i];
                          }
                      cout<<"Your binary number, written as decimal, is: "<<liczb;
                      }
```

Fig. 8. The program source code
*Source: Made by the Author*

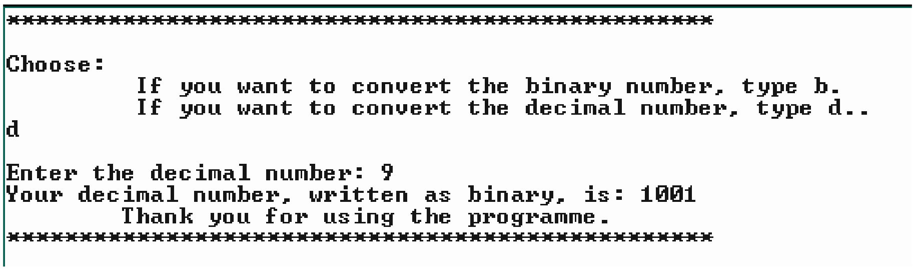
Once it is compiled, the program can be called in a dialog box. Calling a program on the screen is shown in the figure 9, 10 and 11, depends on the letter the user chooses. After choosing the letter b and the binary number 1001, the program will proceed:



```
************************************************************

Choose:
           If you want to convert the binary number, type b.
           If you want to convert the decimal number, type d..
b

Emter the binary number:: 1001
Your binary number, written as decimal, is: 9
         Thank you for using the programme.
************************************************************
```

Fig. 9. Calling the program if user chooses the letter b.
*Source: Made by the Author*

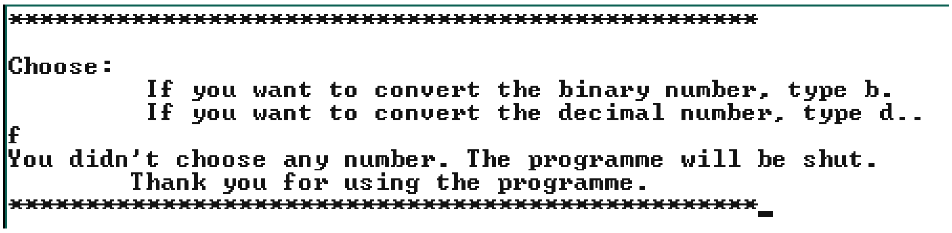
After choosing the letter d and the decimal number 9, the program will proceed:



```
**************************************************

Choose:
         If you want to convert the binary number, type b.
         If you want to convert the decimal number, type d..
d

Enter the decimal number: 9
Your decimal number, written as binary, is: 1001
        Thank you for using the programme.
**************************************************
```

Fig. 10. Calling the program if user chooses the letter d
*Source: Made by the Author*

After choosing the other symbolic or a letter, the program will proceed:



```
**************************************************

Choose:
         If you want to convert the binary number, type b.
         If you want to convert the decimal number, type d..
f
You didn't choose any number. The programme will be shut.
        Thank you for using the programme.
**************************************************_
```

Fig. 11. Calling the program if user chooses the other letter
*Source: Made by the Author*

## 3. Conclusions

- After a short introduction of programming in $C^{++}$, there was proposed the program source code, which easily converts a numbers within both systems.
- To create the program the user will need some basic knowledge of the syntax of $C^{++}$, a wide range of books and courses available in the market.
- This should all be the opportunity to learn more about the $C^{++}$ programming language.
- Conversion of numbers within the two most common numerical systems is widespread, so the ability to create the source code, for example, in $C^{++}$, to convert within the systems, can be very useful.

## References

[1] Grębosz J.: *Symfonia C*$^{++}$, Tom 1, Kraków 1999.

[2] Kisilewicz J.: *Język C*$^{++}$*- programowanie obiektowe*, Oficyna Wyd. Politechniki Wrocławskiej, Wrocław 2005.

[3] Lippman S.B., Lajoie J.: *Podstawy języka C*$^{++}$, Wyd. Naukowo-Techniczne, Warszawa 2000.

[4] Majczak A. K.: *C*$^{++}$ *w 48 godzin*, Warszawa 1993.

[5] Porębski W. M.: *Programowanie w języku C*$^{++}$, Gdańsk 1998.

[6] Soulie J.: *C*$^{++}$*language tutorial*, 2008 *(http://www.cplusplus.com/doc/tutorial/).*