**Converting Program**

**Project Summary**

This program converts numbers from binary to decimal and from decimal to binary.

**Getting Started**

Download Visual C++ 2017 Community: A free Windows C++ compiler by Microsoft.

Download DEV-C++ 5.11

**How to run the program**

Step 1: Configure Dev-C++.

We need to modify one of the default settings to allow you to use the debugger with your programs.

Step 2: Create a new project.

A “project” can be considered as a container that is used to store all the elements that are required to compile a program.

Step 3: Create/add source file(s).

Step 4: Compile.

Once you have entered all of your source code, you are ready to compile.

Step 5: Execute.

You can now run your program.

Step 6: Debug.

When things aren’t happening the way you planned, a source-level debugger can be a great tool in determining what really is going on. Dev-C++’s basic debugger functions are controlled via the “Debug” tab at the bottom of the screen; more advanced functions are available in the “Debug” menu.

**Instruction of the program**

**C++ standard library headers**

|  |  |
| --- | --- |
| **Library headers** | **Description** |
| #include <iostream> | Header that defines the standard input/output stream objects. |
| #include <cmath> | Header that declares a set of functions to compute common mathematical operations and transformations. |
| #include <string> | This header introduces string types, character traits and a set of converting functions. |

**using namespace std;**

are used to define a scope and allows us to group global classes, functions and/or objects into one group.

**Global Variables and Prototypes**

|  |  |
| --- | --- |
| **Global Variables and Prototypes** | **Description** |
| int numbers; | It is a variable numbers that return value as an integer. |
| int option; | It is a variable option that return values as an integer. |
| int binaryToDecimal (int numbers); | binaryToDecimal is a prototype that return values as an integer. |
| void decimalToBinary (int numbers); | decimalToBinary is a void prototype that return values as an integer. |

**int main() function** - This line initiates the declaration of a function.

**In the main function:**

|  |  |
| --- | --- |
| **Print Out Sentences** | **Descriptions** |
| cout << “1) Convert Binary to Decimal.” << endl; | A sentence that tells the user the choice/option one is for converting numbers from binary to decimal. |
| cout << “2) Convert Decimal to Binary.” << endl; | A sentence that tells the user the choice/option number two is for converting numbers from decimal to binary. |
| cout << “Enter 1, 2, or 0 to exit.” << endl; | A sentence that tells the user to enter 0, 1, or 2. The program will be done if the user enters 0. |
| cout << “Choose a number from 0, 1, 2:” ; | A sentence that asks the user to enter one of these numbers’ choices. |
| cin >> option; | The user enters one of the numbers of their choice to either converting numbers or exiting the program. |
| cout << endl; | The purpose of using endl in this case is to make more space to make program looks nicer and organized. |

**Using while loop for an option- while(option)**

**Inside the while loop using-If (option==1)**

|  |  |
| --- | --- |
| **Print out** | **Descriptions** |
| cout <<”1) Binary to Decimal Conversion.” <<endl; | If the user chooses number 1, this sentence will print out. |
| cout << “Enter the binary number(numbers <=1000 000 000):” << endl; | This sentence lets the user knows that they can enter the binary number into the program. The number has to be less than or equal 1000 000 000. If the number is bigger than this, the program will be error and keep repeating the last sentence that print out nonstop. |
| cin >> numbers; | It allows the user to insert the binary number for converting. |
| cout << numbers << “ = “ binaryToDecimal(numbers) | * numbers\_ numbers that the user inserts. * Using “ = “ to let the user knows that the numbers that they insert equals to the result. * BinaryToDecimal(numbers)-function that has the result for converting from binary to decimal. |

**else if (option ==2) if the user enters 2 as their option.**

|  |  |
| --- | --- |
| **Print out** | **Descriptions** |
| cout << “2) Decimal to Binary Conversion.” << endl; | If the user chooses number 2, this sentence will print out. |
| cout << “Enter the decimal numbers <=1000 000 000):”; | This sentence lets the user knows that they can enter the decimal number into the program. The number has to be less than or equal 1000 000 000. If the number is bigger than this, the program will be error and keep repeating the last sentence that print out nonstop. |
| cin >> numbers; | It allows the user to insert the decimal number for converting. |
| decimalToBinary(numbers); | void function that provide the result for converting from decimal to binary. |
| cout << endl; | The purpose of using endl in this case is to make more space for making program looks nicer and organized. |

**Using else, if the user enters other numbers that is not 1, 2, or 0.**

|  |  |
| --- | --- |
| **Print out** | **Description** |
| cout << “Invalid number. Please try again.” | To let the user knows that the option that they insert is not the correct option, so they can insert a number again. |

After if-else if- else. Print out these sentences in the while loop for repeating the process.

|  |
| --- |
| **Print out** |
| cout << "1) Convert Binary to Decimal." << endl; |
| cout << "2) Convert Decimal to Binary." << endl; |
| cout << "Enter 1, 2, or 0 to exit." << endl <<endl; |
| cout << "Choose a number from 0, 1, and 2: "; |
| cin >> option; |
| cout << endl; |

At the end of the main function, we need to write return 0, to return all the values and It is like saying the program works fine.

**Outside the main function, working on two functions individually:**

1. Function one is: **int binaryToDecimal (int numbers)**- converting numbers from binary to decimal and return values as integer.

|  |  |
| --- | --- |
| **int binaryToDecimal (int numbers)** | |
| Declare variables that we need in this function: | int numbers1 = numbers;  int remainder;  int decimal = 0;  int i = 0; |
| using while loop that numbers1 is different from 0. | while (numbers1 != 0)  { } |
| In the while loop, do some formula how to convert from binary to decimal. | Remainder = numbers1 %10;  //finding remainder by modulus numbers1 and 10.  Decimal = decimal + remainder \* pow(2,i);  i++;  //after finding the remainder, multiple it by 2 exponent i (i starts from 0 and keep increasing) and adding it to decimal.  number1 = numbers1 / 10;  //the next step, divide numbers1 by 10.  Repeating the process |
| Return decimal after the while loop | |

1. Function two is: **void decimalToBinary(int numbers)**-converting numbers from decimal to binary and return values as integer.

|  |  |
| --- | --- |
| void decimalToBinary (int numbers) | |
| Declare variables that we need in this function: | int numbers1 = numbers;  int remainder;  string binary = “ ”; |
| Using while loop that numbers1 is different from 0. | Remainder = numbers1 %2;  //finding remainder by modulus numbers1 by 2.  binary = char(remainder + 48) + binary;  //after finding the remainder, using ASCii table by adding the remainder with 48 and change it into character to make result reverse into the correct order.  number1= numbers1 /2;  //next step, divide numbers1 by 2.  Repeating the process. |
| Outside the while loop. | cout << numbers << “=” << binary;  //numbers that user insert equals to the results (binary is the result after converting)  cout << endl;  //endl to make program looks more organized. |