A Level Computer Science Bridging Project

Task: Denary/Binary Calculator

[**Flowchart - First Mockup**](#_m799ynumuld1) **2**

[**Pseudocode - Second Mockup**](#_soemkfdmcmvh) **3**

[**Javascript - Third Mockup**](#_eeevkh6j8579) **4**

[**Final Code**](#_gek9t3unsgn6) **5**

[**Program Annotations**](#_mqiph8k1y32s) **7**

[**Completed Test Plan**](#_uue4gug4sxhg) **8**

[**Evaluation**](#_ufe2psjeepus) **9**

# 

# Criteria:

Design, code, test and evaluate a program in JavaScript that will convert between binary and decimal. For the decimal to binary converter the program should accept a positive value and output the binary equivalent. The system need only be tested for values up to 255. It should also have a basic output for user misuse and any unexpected inputs.

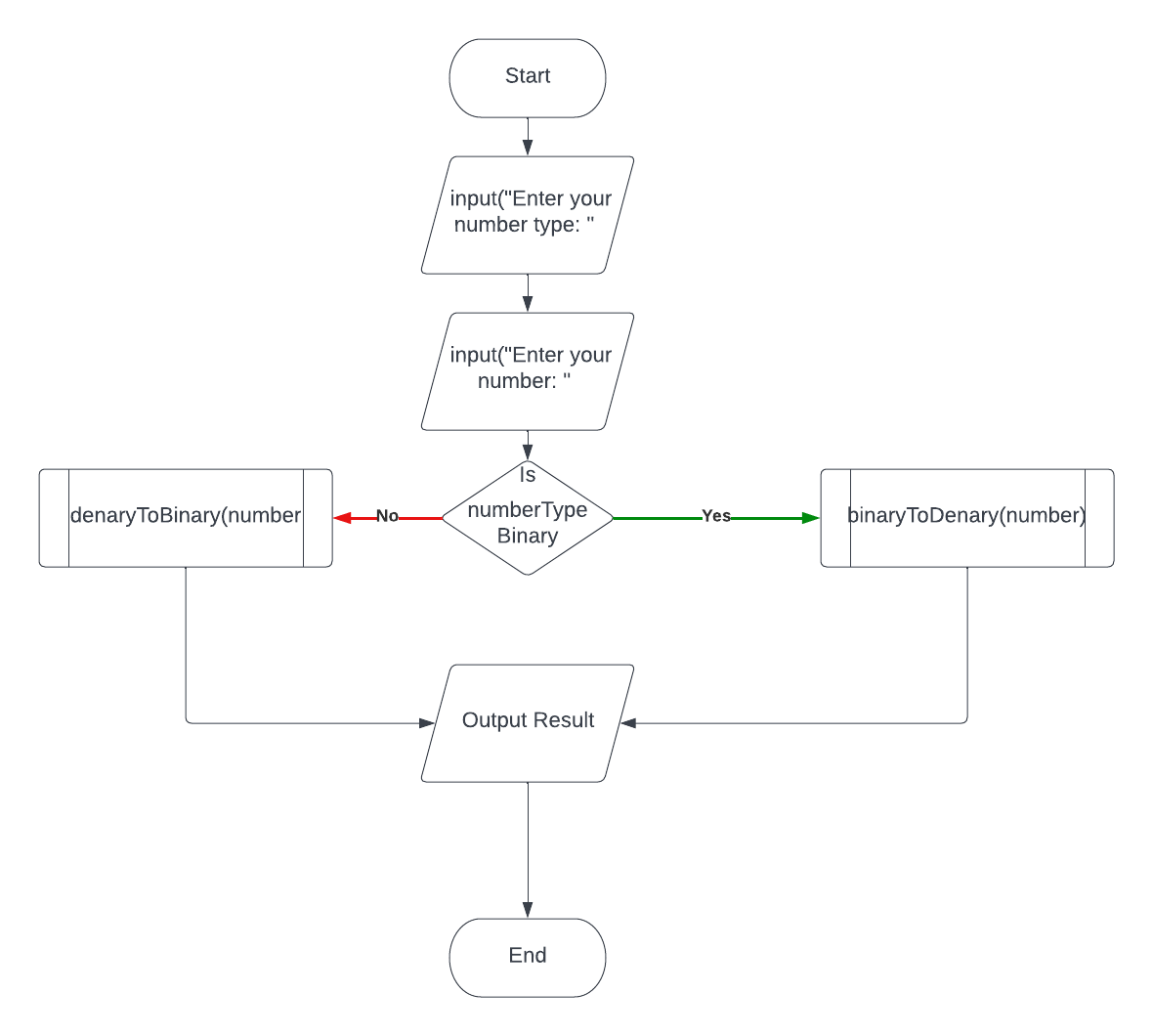
# 

# 

# 

I used <https://playcode.io/> to write my code

# Flowchart - First Mockup



# Pseudocode - Second Mockup

input numberType = input("Is it denary or binary").lower()

input number = int(input("Enter your number: "))

input remainder = 0

input integerRemainder = 0

input binaryNumber = []

integerRemainder = number

input characters = []

input decimalTotal = 0

input loopCount = 0

if number Type == "denary" Then

While integerRemainder != 0 Do

remainder = integerRemainder % 2

binaryNumber.Add(remainder)

integerRemainder = integerRemainder/2.Round()

print(number + " is" + binaryNumber + " as a binary number")

ElseIf numberType == "binary" Then

For i = (number.Len-1), i >= 0, i-- Do

decimalTotal = decimalTotal + number[i] \* 1 \*(2\*\*loopCount)

loopCount ++

print(number + " is" + decimalTotal + " as a denary number")

Else Do

print("Please enter valid entries.")

# Javascript - Third Mockup

*//Accepts the number and type from the user*

let numberType = prompt("Is your number a denary or binary?: ").toLowerCase();

let number = prompt("Enter your number: ");

*//Variables for decimal to binary*

let remainder;

let integerRemainder;

let binaryNumber = new Array;

integerRemainder = number;

*//Variables for binary to decimal*

let characters = new Array();

for (var i = 0; i < number.length ; i++)

{

characters[i] = number[i]

}

console.log(characters)

*//Decides what the given number is*

if (numberType == "denary") {

*// Repetes through the loop until the remainder of the integer division is 0*

while (integerRemainder != 0 ){

*//Calculates the remainder of the division*

remainder = integerRemainder % 2;

*//Appends the remainder(e.g.Binary NUmber) to the array*

binaryNumber.push(remainder);

*//Carries out integer division*

integerRemainder = Math.floor(integerRemainder/2);

}

*//Outputs the final binary number as a string*

console.log(number + " is " + binaryNumber.join() + "As A Binary Number")

} (binaryNumber)

# Final Code

*//Accepts the number and type from the user*

let numberType;

let number;

*//Variables for decimal to binary*

let remainder;

let integerRemainder;

let binaryNumber = [];

integerRemainder = number;

*//Variables for binary to decimal*

let characters = [];

let decimalTotal;

let loopCount = 0;

numberType = window.prompt("Is your number a denary or binary?: ").toLowerCase();

number = prompt("Enter your number: ");

*//Variables for decimal to binary*

remainder;

integerRemainder;

binaryNumber = [];

integerRemainder = number;

*//Variables for binary to decimal*

decimalTotal = 0

loopCount = 0;

*//Decides what the given number is*

if (numberType == "denary") {

*//-----Decimal Number-----*

*// Repetes through the loop until the remainder of the integer division is 0*

while (integerRemainder != 0 ){

*//Calculates the remainder of the division*

remainder = integerRemainder % 2;

*//Appends the remainder(e.g.Binary NUmber) to the array*

binaryNumber.unshift(remainder);

*//Carries out integer division*

integerRemainder = Math.floor(integerRemainder/2);

}

*//Outputs the final binary number as a string*

console.log(number + " is " + binaryNumber.join().replaceAll(",","") + " As A Binary Number");

}

else if (numberType == "binary") {

*//-----Binary Number-----*

*//Loops through each digit in the number*

for (let i = (number.length-1); i >= 0 ; i--){

*//Multiplies the current digit by 2 to the power of the current loop count*

decimalTotal = decimalTotal + number[i] \*1 \*( 2 \*\* loopCount);

*//Increments the loop count by 1*

loopCount++;

}

*//Outputs the denary number*

console.log(number + " Is "+ decimalTotal + " As A Denary Number");

}

else {

console.log("Please enter valid entries.")

}

# Program Annotations

Binary - Denary



Denary - Binary



# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Completed Test Plan

| Test Case Type | Description | Test | Expected Result | Actual Result |
| --- | --- | --- | --- | --- |
| Functionality | The program should output accurate results from 0-255 | BINARY TO DENARY:  input numbers:  00000001  00001111  11111111  DENARY TO BINARY:  input numbers:  001,11.5,100,255 | BINARY TO DENARY:  00000001 = 1  00001111 = 15  11111111 = 255  DENARY TO BINARY:  001 = 00000001  11.5= 00001011.5  100 = 01100100  255 = 11111111 |  |
| Anticipating Misuse | The program should be able to anticipate user misuse and continue to function | Invalid inputs, such as “blue”,”Granite”, will be entered in the first and second entries. | The program should give a response, such as “Please enter valid inputs.” |  |
|  | | | | |
| Functionality Results |  | | |  |

# Evaluation

From my testing, the program performed correctly, it gave accurate answers for all inputs. The only functional thing I would add would be to make the conversion code for each one a subroutine so it could be easily repeated for as many numbers as the user wants to input without having to restart the program each time. Any other things i would add would be purely cosmetic, for example, when an output is given in binary, it should be given in 8 characters for anything below and including 255, the current program does not add any extra 0s, this would just add a bit more continuity between all the outputs. Moreover, a dedicated webpage would be greatly beneficial considering the only way to input is through the console or the browsers built in input system. By using a dedicated input box, it would also allow for a lot more web design which would make the calculator seem more user friendly. As Well as allowing the outputs to be easily identified since again the only way to see the output is through the console.

The program also correctly identified unexpected inputs when the user inputs the type of number they are entering, but it failed to identify unexpected inputs when the actual number was being imputed, this led to errors being outputted. This can be solved by making sure the input is an integer. Additionally, perhaps adding more detailed explanations about why the program rejected the users input could be useful, for example “Please enter a valid number”, rather than the same explanation no matter the reason.

The next big addition would be to add other conversion options such as hexadecimal and allow for calculations (addition,multiplication,etc) to be made in one number system and then to outputted in another

In summary, my program meets the criteria with 1 small issue which can be easily resolved. The actual program itself is an easy to understand program with many comments explaining what the code does, as well as lots of white space breaking up the chunks of code.