

202151160_Lab5

Name: Snehal Keshav Nalawade
ID: 202151160

1)

Objective: To write a program to convert a decimal number into its binary equivalent

Software: Online GDB Compiler and Debugger for C (IDE)

Methodology:

To convert a decimal number into its binary equivalent, Divide the given decimal number by 2 and note down the remainder. Now, divide the obtained quotient by 2, and note the remainder again. Repeat the above steps until you get 0 as the quotient. Now, write the remainders in such a way that the last remainder is written first, followed by the rest in the reverse order.

Algorithm:

Step 1) Start

Step 2) Declare and initialize variables

Step 3) Read the input and store it in a variable

Step 4) If input > 0, enter the for loop and execute step 5 and 6

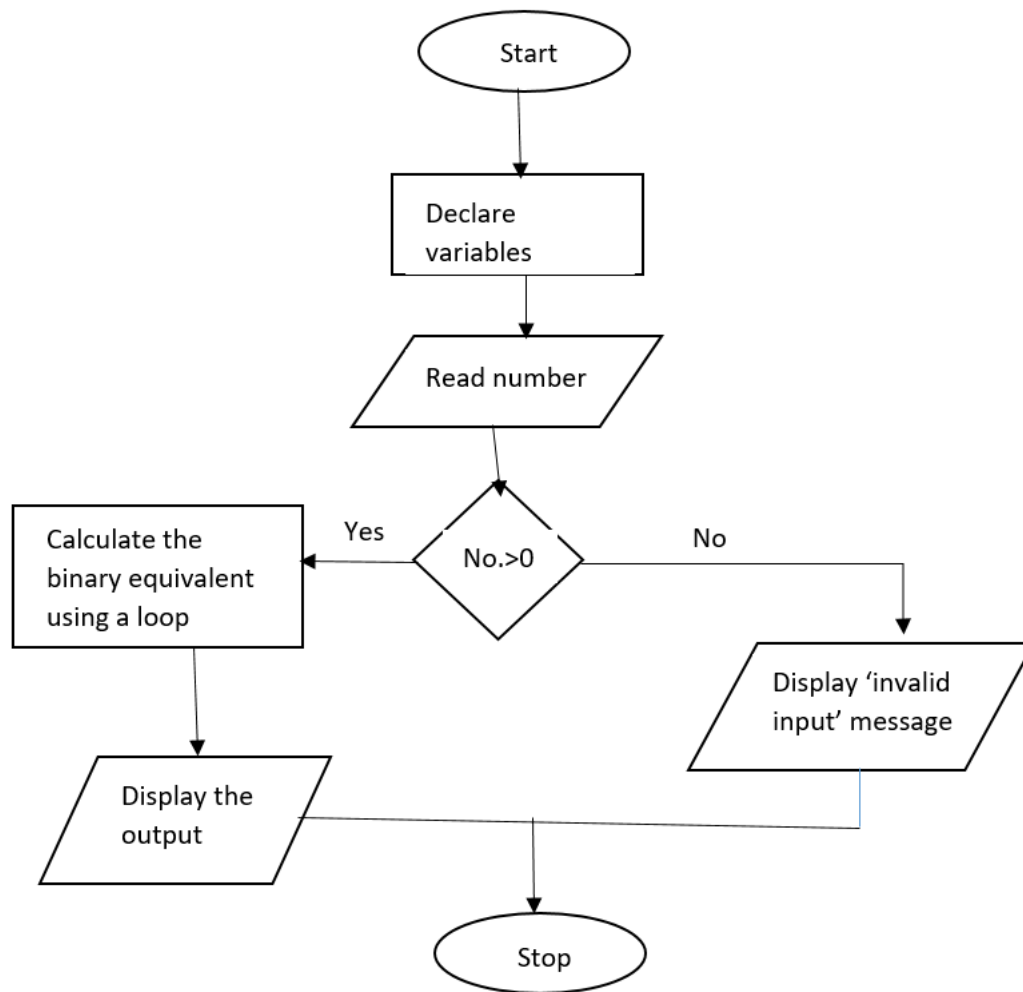
Step 5) calculate the binary equivalent and print output after each loop cycle

Step 6) display the final result i.e. the binary equivalent

Step 7) if input ≤ 0, the compiler will display a message of invalid input

Step 8) Stop

Flowchart:



Code:

**/* This C program is prepared by Snehal Keshav Nalawade
ID : 202151160
Date of preparation : 25/01/2022**

This program reads a positive integer number from the user and displays it's binary equivalent.

The result is achieved by using for loop.

***/**

```
#include<stdio.h>    // importing std input, output header files
int main(void)
{
    int a, num, rem, binary_num=0, c=1, b=1;
    printf("Enter a Number : ");
    scanf("%d",&a);    // taking input from the user
    num=a;
    for(;num>0;num/=2)
    {
        // start of the for loop
        rem=num%2;
```

```

        binary_num+=(rem*c);
        printf("\n\nStep\n%d:\t%d/2,\tRemainder=%d,\tQuotient=%d",b,num,rem,(num/2));
        b++;          // increment operator used for variable b
        c*=10;
    }
    if(a>0)
        printf("\n\n%d in decimal = %d in binary",a,binary_num);
    else
        printf("Invalid input: Enter A positive Integer only");
    return 0;
}          // closing of the main function

```

Sample Output:

The screenshot displays the OnlineGDB beta interface. On the left is a sidebar with navigation links: 'Welcome, Snehal Nalawade', 'lab5 decimaltobinary', 'Create New Project', 'My Projects', 'Classroom new', 'Learn Programming', 'Programming Questions', 'Logout', and social media icons. The main area shows the program's execution steps:

Step	Operation	Remainder	Quotient
1:	354/2,	0	177
2:	177/2,	1	88
3:	88/2,	0	44
4:	44/2,	0	22
5:	22/2,	0	11
6:	11/2,	1	5
7:	5/2,	1	2
8:	2/2,	0	1
9:	1/2,	1	0

Below the steps, the final result is shown: "354 in decimal = 101100010 in binary". The program finished with exit code 0, and the console prompt "Press ENTER to exit console." is visible.

Conclusion: The C code has been executed successfully and the desired results are obtained.

2)

Objective: To write a program to convert a binary number to its decimal equivalent

Software: Online GDB Compiler and Debugger for C (IDE)

Methodology:

To convert a binary number to its decimal equivalent, list out the powers of 2 for all the digits starting from the rightmost position. The first power would be 2^0 and as we move on it will be 2^1 , 2^2 , 2^3 , 2^4 , 2^5

Algorithm:

Step 1) Start

Step 2) Declare variables

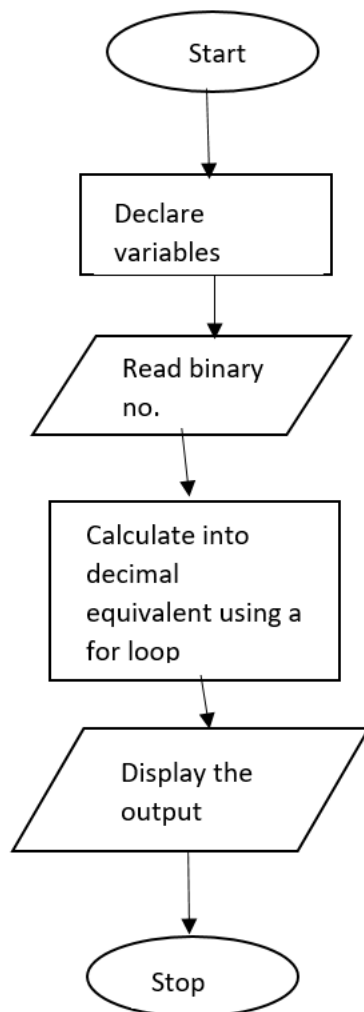
Step 3) Read a binary number from the user

Step 4) use for loop to calculate its decimal equivalent,
Method of sum of powers of 2 to be used

Step 5) display the result

Step 6) Stop

Flowchart:



Code:

**/* This C program is prepared by Snehal Keshav Nalawade
ID : 202151160
Date of preparation : 25/01/2022**

This program reads a binary number from the user, calculates its decimal equivalent and displays the entire result.

the program is written using for loop and method of sum of powers of 2

***/**

```
#include<stdio.h>
int main(void)
{
    long int binary_num, B;      // declaration of variables
    int i=1, rem, num=0, p=0, c=1;
    printf("Enter a binary number: ");
    scanf("%ld",&binary_num);    // reading input from the user
    B=binary_num;
```

```

for(;B>0;)
{
    i=p+1;    // since 2^n=(sum of first (n-1) powers of 2)+1
    rem=B%10;
    num+=(rem*i); // adding up the decimal equivalents of binary digits
    p+=i;
    B/=10;
    printf("\nStep\n%d: Remainder=%d, Left code=%ld, calculated value=%d",c,rem,B,num);
    c++;
}
printf("\n\n\t%d in binary = %d in decimal",binary_num,num);    // displaying the final result
return 0;
}    // closing of the main function

```

Sample Output :

The screenshot shows the OnlineGDB beta interface. On the left is a sidebar with navigation links: Welcome, Snehal Nalawade; lab5 binarytodecimal; Create New Project; My Projects; Classroom (new); Learn Programming; Programming Questions; Logout; and social media icons for Facebook, Twitter, and a plus sign. The main area displays the execution of a C program. At the top, there's a toolbar with icons for file operations, Run, Debug, Stop, Share, Save, Beautify, and a download icon. Below the toolbar, the file name 'main.c' is shown. The input field contains '101100010'. The output area shows the following steps:

```

Enter a binary number: 101100010
Step
1: Remainder=0, Left code=10110001, calculated value=0
Step
2: Remainder=1, Left code=1011000, calculated value=2
Step
3: Remainder=0, Left code=101100, calculated value=2
Step
4: Remainder=0, Left code=10110, calculated value=2
Step
5: Remainder=0, Left code=1011, calculated value=2
Step
6: Remainder=1, Left code=101, calculated value=34
Step
7: Remainder=1, Left code=10, calculated value=98
Step
8: Remainder=0, Left code=1, calculated value=98
Step
9: Remainder=1, Left code=0, calculated value=354

101100010 in binary = 354 in decimal

...Program finished with exit code 0
Press ENTER to exit console.

```

Conclusion: The C code has been executed successfully and the desired results are obtained.

Thank you