**Manan Gandhi**

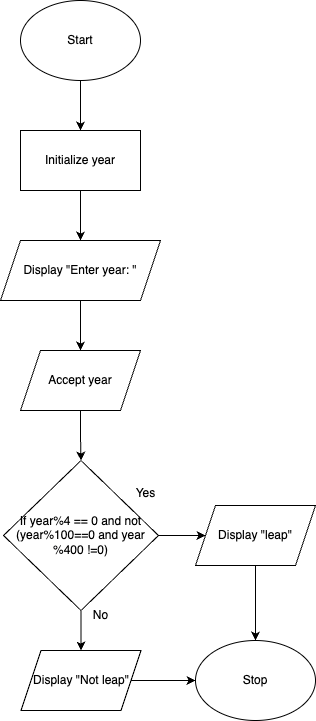
**C146 – D1 – Exp2 Part C**

**Q1.** Implement a code to identify whether the input year is leap year or not

**Algorithm:**

1. Start
2. Initialize year
3. Display “Enter year: ”
4. Accept year
5. If year% 4 ==0 and not (year%100==0 and year%400!=0), goto 6 else goto 8
6. Display “Leap”
7. Goto 9
8. Display “Not leap”
9. Stop

**Flowchart:**



**Code:**

#include <iostream>

using namespace std;

int main()

{

int year;

cout << "Enter year: ";

cin >> year;

if (year % 4 == 0 && !(year % 100 == 0 && year % 400 != 0))

{

cout << year << " is a leap year" << endl;

}

else

{

cout << year << " is not a leap year" << endl;

}

return 0;

}

**Output:**

A close up of white text

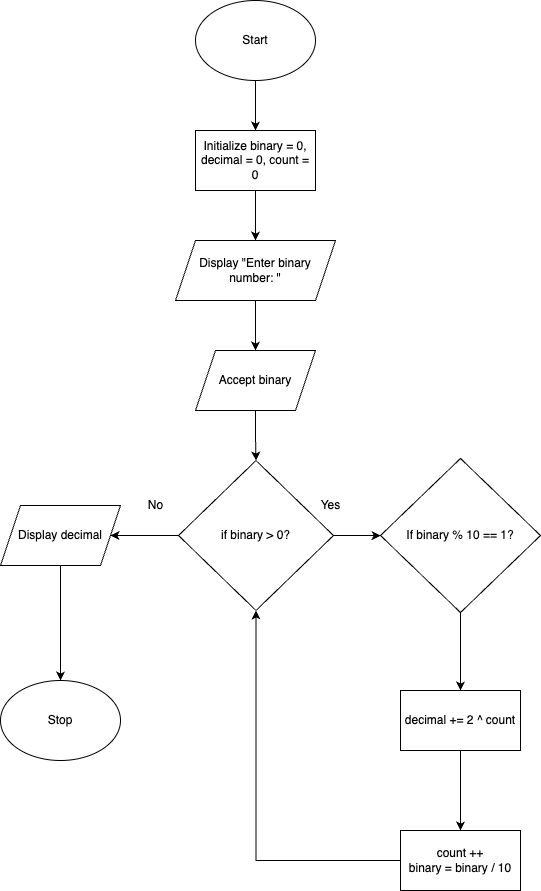
Description automatically generated

**Q2.** Implement a code to convert binary to decimal equivalent

**Algorithm:**

1. Start
2. Initialize binary = 0, decimal = 0, count = 0
3. Display “Enter binary number: ”
4. Accept binary
5. If binary > 0, goto 6 else goto 10
6. If binary % 10 == 1, decimal = 2^count
7. Count ++
8. Binary = binary / 10
9. Goto 5
10. Display decimal
11. Stop

**Flowchart:**



**Code:**

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

int binary = 0, decimal = 0;

cout << "Enter binary number: ";

cin >> binary;

int count = 0;

while (binary > 0)

{

if (binary % 10 == 1)

{

decimal += pow(2, count);

}

count++;

binary /= 10;

}

cout << decimal << endl;

return 0;

}

**Output:**

A blue background with white text

Description automatically generated

**Q3.** Implement a program to find GCD of N numbers

**Algorithm:**

1. **Start**
2. **Initialize a and b with the two numbers**
3. **Display "Enter two numbers: "**
4. **Accept a and b**
5. **If b == 0, goto 9 else goto 6**
6. **Set remainder = a%b**
7. **Set a = b**
8. **Set b = remainder, goto 5**
9. **Display GCD, which is a**
10. **Stop**

**Code:**

#include <iostream>

using namespace std;

int main()

{

int a, b;

cout << "Enter a: ";

cin >> a;

cout << "Enter b: ";

cin >> b;

while (a != b)

{

if (a > b)

{

a -= b;

}

else

{

b -= a;

}

}

cout << a << endl;

return 0;

}

**Output:**

A blue background with white letters

Description automatically generated

**Q4.** Write a program to find the roots of a quadratic equation

**Algorithm:**

1. Start
2. Initialize a, b, c, r1, r2
3. Display “Enter a, b, c: ”
4. Accept a, b, c
5. R1 = -b + sqrt(pow(b, 2) – 4\*a\*c)
6. R2 = -b - sqrt(pow(b, 2) – 4\*a\*c)
7. Display r1, r2
8. Stop

**Code:**

#include <iostream>

using namespace std;

int main()

{

int a, b, c;

cout << "Enter a, b, c in the following format ax^2 + bx + c: "<<endl;

cout << "a: ";

cin >> a;

cout << "b: ";

cin >> b;

cout << "c: ";

cin >> c;

float r1 = -b + sqrt(pow(b, 2) - 4 \* a \* c);

float r2 = -b - sqrt(pow(b, 2) - 4 \* a \* c);

cout << r1 << endl;

cout << r2 << endl;

return 0;

}

**Output:**

**A screen shot of a computer

Description automatically generated**

**Q5.** Implement a program to shutdown windows/linux system

**Algorithm:**

1. Start
2. System(“shutdown /s”)
3. Stop

**Code:**

#include <iostream>

using namespace std;

int main()

{

system("shutdown /s");

return 0;

}