/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* This function will fill an array with the decimal equivalent \*\*

\*\* of 1/2, 1/4, 1/8, 1/16 .... \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fillArray(double arr[])

{

double den = 2.0;

for (int i = 0; i < SIZE; i++)

{

arr[i] = 1.0/den;

den \*= 2.0;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* This function will convert a whole number into a binary \*\*

\*\* number by performing successive division and saving the \*\*

\*\* remainders in a sting \*\*

\*\* Note: requires reversing the resultant string \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void convertWhole(int num, string &s)

{

stringstream out;

int remainder;

while (num > 0)

{

remainder = num % 2;

out << remainder; //send remainder to the stingstream

num /= 2;

}

s = out.str(); //send the stringstream to the string

//Reverse the string

s = reverseString(s);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* This function reverses a string that is given to it \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

string reverseString(string s)

{

string temp = "";

for (int i = s.length() - 1; i >= 0; i--)

{

temp += s[i];

}

return temp;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* This function will generate the bit sequence for the \*\*

\*\* fractional part of a decimal number \*\*

\*\* \*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void convertFraction(double num, string &s, double arr[])

{

// num contains the fractional part of a number

// &s is a reference parameter to the original string

// arr[] contains successively smaller powers of 1/2

for (int i = 0; i < SIZE; i++)

{

if (num >= arr[i]) //add a 1 to the string

{

s += "1";

num -= arr[i]; //reduce num by the amount accounted

//for by the one in place

}

else

s += "0"; //add a 0 to the string

}

}