#include<iostream>

#include<cstdlib>

#include<cmath>

#define SIZE 8

using namespace std;

//Function declarations/prototypes!

void generate\_binary(int[]);

void generate\_hexidecimal(int[], int[]);

void print\_hexidecimal(int []);

void print\_binary( int[]);

//Main function to execute program:

int main(){

//Declare variable and variable holders:

int binary[8]={0,0,0,0,0,0,0,0};

int hexidecimal[2]={0,0};

int decimal = 0;

//Print each initial value of zero to screen

cout<< decimal;

print\_binary(binary);

print\_hexidecimal(hexidecimal);

cout<<"\n";

//Loop to generate numbers from 0-255

while(++decimal < 256)

{

//Print each value and use functions to determine binary and hexidecimal #'s for each decimal #;

cout<<decimal;

generate\_binary(binary);

print\_binary(binary);

generate\_hexidecimal(binary, hexidecimal);

print\_hexidecimal(hexidecimal);

hexidecimal[0]=hexidecimal[1]=0;

cout<<"\n";

}

cout<<" "<< cout<<"/n";

//End of Program

return 0;

}

void generate\_binary(int binary[])

{

int i;

for(i = 7; i > 0; i--){

if(binary[i]==0){

binary[i]=1;

break;

}

else{

binary[i]=0;

}

}

}

void generate\_hexidecimal(int binary[], int hexidecimal[])

{

int i;

for(i = 7; i > 0; i--)

{

if(i>=4)

{

if(binary[i]==1){

hexidecimal[1]+= pow(2,7-i);

}

}

else{

if(i >=3)

{

if(binary[i]==1){

hexidecimal[0]+= pow(2,3-i);

}

}

}

}

}

void print\_hexidecimal(int hexidecimal[])

{ //Print Function if the decimal # is past 9 and less the 16 we output it's hexidecimal letter representation;

int i;

for(i = 0; i < 2; i++){

if(hexidecimal[i]==10){

cout<<"A"<<endl;

}

else if(hexidecimal[i]==11){

cout<<"B"<<endl;

}

else if(hexidecimal[i]==12){

cout<<"C"<<endl;

}

else if(hexidecimal[i]==13){

}

else if(hexidecimal[i]==14){

cout<<"E"<<endl;

}

else if(hexidecimal[i]==15){

cout<<"F"<<endl;

}else{

cout<<hexidecimal[i];

}

}

cout<<"/n";

}

void print\_binary( int binary[])

{

int i;

for(i = 0; i < 7; i++)

{

cout<< binary[i];

}

cout<<"\n";

}