

Program code and output for binary.c

Chris Fariss *

Program Code for binary.c

The program first, displays the decimal, binary and hexadecimal numbers for all values possible in the first digit of a hexadecimal number. Next, calculate the binary version of user inputted integer between 0 and 1048575. The C code %d displays the integer as it was entered. The C code %x displays the integer in its hexadecimal equivalent. the if-then-else statements and while loops create the sequence of binary numbers. */

```
#include<stdio.h>

int main(void){

/* declare a vector to hold up to five hexadecimal digits. */
    int temp[5];

/* declared loop counter variables. */
    int dec1, count, i;

/* declare four digits of a binary number. */
    int b1, b2, b3, b4;

/* declare variables used to calculate the hexadecimal equivalent of a user inputted integer. */
    int chex1, chex2, chex3, chex4, chex5;

/* Display the headers for the three columns that will hold the same numbers in decimal, binary and hexadecimal form.
*/
    printf("Dec Binary Hex\n");

/* Initialize variable to 0 to iterate through the while loop*/
    dec1=0;

/* 4 nested while loops that iteratate through the binary numbers equivalent to the integer value of dec1.*/
```

*This program was created at UCSD during Keith Poole's POLI 279: C by the Beach class.

```

while(dec1 < 16){

/* set the 4th binary digit */
    b4=0;
    while(b4<2){

/* set the 3rd binary digit */
        b3=0;
        while(b3<2){

/* set the 2nd binary digit */
            b2=0;
            while(b2<2){

/* set the 1st binary digit */
                b1=0;
                while(b1<2){

/* prints to screen the decimal, binary and hexidecimal form of the integer variable dec1. */
                    printf(" %2d  %d%d%d%d %2x\n",dec1, b4, b3, b2, b1, dec1);

/* increase the decimal digit and exit the innermost loop */
                    dec1=dec1 + 1;

/* increase the binary digit before exiting each loop beginning with the innermost loop (1st binary digit) */
                    b1=b1+1;
                }

                b2=b2+1;
            }

            b3=b3+1;
        }

        b4=b4+1;
    }

}

/* Get user input, display user input, use C command %x to convert user input into hexadecimal, and display the user
input in binary form */
puts("\nEnter an integer between 0 and 1048575:"); /* user prompt */
scanf ("%d", &chex1);
printf ( "\nThe integer is: %d\n", chex1);
printf("The integer converted into hexadecimal is:
printf("The integer converted into binary is: ");

```

```

/* the if-then-else statements below determines:
(1) the number of hexadecimal digits of the user inputted integer
(2) sets the binary loop count variable based on the number of hexadecimal digits
(3) calculates the integer value of each hexadecimal digit and stores the value in 1 of 5 chex variables using integer
division
(4) stores the value of each hexadecimal digit (1 of 5 chex variables) in the temp vector for use in the binary loop below
*/

```

```

/* Calculate a 1-digit hexadecimal number */
/* the highest decimal value is 15 */

```

```

if(chex1<16){
    count=1;
    temp[0]=chex1;
}

```

```

/* or calculate a 2-digit hexadecimal number */
/* the highest decimal value is 255 */

```

```

else if(chex1<256){
    count=2;
    chex2 = (chex1/16);
    chex1 = chex1-(chex2*16);

    temp[1]=chex1;
    temp[0]=chex2;
}

```

```

/* or calculate a 3-digit hexadecimal number */
/* the highest decimal value is 4095 */

```

```

else if(chex1<4096){
    count=3;
    chex3 = (chex1/256);
    chex2 = chex1-(chex3*256);
    chex2 = (chex2/16);
    chex1 = chex1-(chex3*256)-(chex2*16);

    temp[2]=chex1;
    temp[1]=chex2;
    temp[0]=chex3;
}

```

```

/* or calculate a 4-digit hexadecimal number */
/* the highest decimal value is 65535 */

```

```

else if(chex1<65536){
    count=4;

```

```

chex4 = (chex1/4096);
chex3 = chex1-(chex4*4096);
chex3 = (chex3/256);
chex2 = chex1-(chex4*4096)-(chex3*256);
chex2 = (chex2/16);
chex1 = chex1-(chex4*4096)-(chex3*256)-(chex2*16);

temp[3]=chex1;
temp[2]=chex2;
temp[1]=chex3;
temp[0]=chex4;
}

```

```

/*or calculate a 5-digit hexadecimal number */
/* the highest decimal value is 1048576 */

```

```

else{
count=5;
chex5 = (chex1/65536);
chex4 = chex1-(chex5*65536);
chex4 = (chex4/4096);
chex3 = chex1-(chex5*65536)-(chex4*4096);
chex3 = (chex3/256);
chex2 = chex1-(chex5*65536)-(chex4*4096)-(chex3*256);
chex2 = (chex2/16);
chex1 = chex1-(chex5*65536)-(chex4*4096)-(chex3*256)-(chex2*16);

temp[4]=chex1;
temp[3]=chex2;
temp[2]=chex3;
temp[1]=chex4;
temp[0]=chex5;
}

```

```

/* initializes the value to iterate through the outermost while loop to 0*/

```

```

i=0;

```

```

/* the number of while loop iterations are determined by the number of hexadecimal digits and set to the variable count
*/

```

```

while(i < count){
dec1=0;
while(dec1 < 16){

```

```

/* set the 4th binary digit */

```

```

b4=0;
while(b4<2){

```

```

/* set the 3rd binary digit */

```

```

    b3=0;
    while(b3<2){

/* set the 2nd binary digit */

    b2=0;
    while(b2<2){

/* set the 1st binary digit */

    b1=0;
    while(b1<2){

/* Tests for equivalence between the dec1 variable and the value in the vector temp (as determined by i). The loop
continues to iterate without printing to the screen until these values are equal. The binary value is computed in the
same way for each hexadecimal digit, which are contained in the vector temp. */

        if(temp[i]==dec1){

/* If the test==TRUE then print to screen the binary form variable dec1. */
/* then increase the decimal digit and exit the innermost loop */
/* and increase the decimal digit and exit the innermost loop */

            printf("%d%d%d%d",b4, b3, b2, b1);
            dec1=dec1 + 1;
            b1=b1+1;
        }

/* or if the test==FALSE then just increase the decimal digit and exit the innermost loop */
/* and increase the decimal digit and exit the innermost loop */

        else{
            dec1=dec1 + 1;
            b1=b1+1;
        }
    }

    b2=b2+1;
}

    b3=b3+1;
}

    b4=b4+1;
}

}

/* iterate to the beginning of the outermost while loop or to the next lowest hexadecimal digit */

```

```
i=i+1;
}
```

/ verify that the values contained in the vector match the original hexadecimal number as determined by the previous nested while loop*/*

```
printf("\n");
printf("\ncheck calculation of each hex digit (in its decimal equivalent): %d", count);
if(i==1){
printf("\n hex digit 1 = %d", temp[0]);
}
else if(i==2){
printf("\n hex digit 2 = %d", temp[0]);
printf("\n hex digit 1 = %d", temp[1]);
}
else if(i==3){
printf("\n hex digit 3 = %d", temp[0]);
printf("\n hex digit 2 = %d", temp[1]);
printf("\n hex digit 1 = %d", temp[2]);
}
else if(i==4){
printf("\n hex digit 4 = %d", temp[0]);
printf("\n hex digit 3 = %d", temp[1]);
printf("\n hex digit 2 = %d", temp[2]);
printf("\n hex digit 1 = %d", temp[3]);
}
else{
printf("\n hex digit 5 = %d", temp[0]);
printf("\n hex digit 4 = %d", temp[1]);
printf("\n hex digit 3 = %d", temp[2]);
printf("\n hex digit 2 = %d", temp[3]);
printf("\n hex digit 1 = %d", temp[4]);
}

printf("\n\n");

return(0);
}
```

Program Output for binary.c

1-digit example

```
\singlespace
Dec Binary Hex
 0 0000 0
 1 0001 1
 2 0010 2
 3 0011 3
 4 0100 4
 5 0101 5
 6 0110 6
 7 0111 7
 8 1000 8
 9 1001 9
10 1010 a
11 1011 b
12 1100 c
13 1101 d
14 1110 e
15 1111 f
```

Enter an integer between 0 and 1048575:

12

The integer is: 12

The integer converted into hexadecimal is: c

The integer converted into binary is: 1100

check calculation of each hex digit (in its decimal equivalent): 1

hex digit 1 = 12

logout

2-digit example

Dec	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	a
11	1011	b
12	1100	c
13	1101	d
14	1110	e
15	1111	f

Enter an integer between 0 and 1048575:

189

The integer is: 189

The integer converted into hexadecimal is: bd

The integer converted into binary is: 10111101

check calculation of each hex digit (in its decimal equivalent): 2

hex digit 2 = 11

hex digit 1 = 13

logout

3-digit example

Dec	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	a
11	1011	b
12	1100	c
13	1101	d
14	1110	e
15	1111	f

Enter an integer between 0 and 1048575:

2457

The integer is: 2457

The integer converted into hexadecimal is: 999

The integer converted into binary is: 100110011001

check calculation of each hex digit (in its decimal equivalent): 3

hex digit 3 = 9

hex digit 2 = 9

hex digit 1 = 9

logout

4-digit example

Dec	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	a
11	1011	b
12	1100	c
13	1101	d
14	1110	e
15	1111	f

Enter an integer between 0 and 1048575:

45987

The integer is: 45987

The integer converted into hexadecimal is: b3a3

The integer converted into binary is: 1011001110100011

check calculation of each hex digit (in its decimal equivalent): 4

hex digit 4 = 11

hex digit 3 = 3

hex digit 2 = 10

hex digit 1 = 3

logout

5-digit example

Dec	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	a
11	1011	b
12	1100	c
13	1101	d
14	1110	e
15	1111	f

Enter an integer between 0 and 1048575:

72345

The integer is: 72345

The integer converted into hexadecimal is: 11a99

The integer converted into binary is: 00010001101010011001

check calculation of each hex digit (in its decimal equivalent): 5

hex digit 5 = 1

hex digit 4 = 1

hex digit 3 = 10

hex digit 2 = 9

hex digit 1 = 9

logout