```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>


int main(){


int op;
char secretCode='A';//has a default value of 'A'
//these b1-b8 represents an 8 bit integer. It is equivalent the binary equivalent of 'A',which is 65

int b8=0,b7=1,b6=0,b5=0,b4=0,b3=0,b2=0,b1=1;//these values can be modified If the user selects option1 and changes the secret code.
int ecB8=0,ecB7=0,ecB6=0,ecB5=0,ecB4=0,ecB3=0,ecB2=0,ecB1=0;//ecB stands for "encrypted bit"
int hexToB8=0,hexToB7=1,hexToB6=0,hexToB5=0,hexToB4=0,hexToB3=0,hexToB2=0,hexToB1=1;//hexToB stands for "hexadecimal to binary bit. This is used in option2-base16"
int base;//this variable used in the 2nd option as a base choice.


do{
    printf("\nHappy Encryption!\n");
    printf("(1)Change the secret code\n");
    printf("(2)Encrypt text\n");
    printf("(3)Quit\n");
    printf("You choose: ");
    scanf("%d",&op);
    while(op<1 || op>3){
        fflush(stdin);
        printf("Please enter an option between 1-3!\n");
        printf("You choose: ");
        scanf("%d",&op);
    }

    switch(op){

case 1:

    fflush(stdin);//clean the buffer before requesting for character input
    printf("You have chosen option 1!\n");
    printf("Which secret code will you use? ");//keep in mind that you must only enter a single character
    scanf("%c",&secretCode);
    while((secretCode<48||secretCode>57) && (secretCode<65||secretCode>90) && (secretCode<97||secretCode>122)){
        fflush(stdin);//clean the buffer before requesting for character input
        printf("\nERROR! Secret code can be either:\n");
        printf("A digit between 0-9\n");
        printf("Uppercase letters\n");
        printf("Lowercase characters\n\n");
        printf("Which secret code will you use? ");
        scanf("%c",&secretCode);
    }

    int rem;//this variable is the remainder when we take modulus 2 of the number
    int number=secretCode;//this hold the integer value of the secret code
    int binaryNumber=0;// this is our final conversion result
    int mult=1;//this is for multiplier
    int count=0;//this is the count of binary number elements

    while(number>0){
        rem=number%2;
        binaryNumber=binaryNumber+(rem*mult);
        mult*=10;
        number/=2;
        count++;
    }
```

```
64
65      //as we cant use arrays, I will modify every bit till the number becomes zero
66
67      //this means that the bit8 must remains zero, and I need to modify the bit1 to bit7
68      if(count==7){
69
70          b1=binaryNumber%10;
71          binaryNumber/=10;
72          b2=binaryNumber%10;
73          binaryNumber/=10;
74          b3=binaryNumber%10;
75          binaryNumber/=10;
76          b4=binaryNumber%10;
77          binaryNumber/=10;
78          b5=binaryNumber%10;
79          binaryNumber/=10;
80          b6=binaryNumber%10;
81          binaryNumber/=10;
82          b7=binaryNumber%10;
83          binaryNumber/=10;
84          b8=0;
85      }
86      //this means that the bit7 and 8 must remains zero, and I need to modify the bit1 to bit6
87      else{
88
89          b1=binaryNumber%10;
90          binaryNumber/=10;
91          b2=binaryNumber%10;
92          binaryNumber/=10;
93          b3=binaryNumber%10;
94          binaryNumber/=10;
95          b4=binaryNumber%10;
96          binaryNumber/=10;
97          b5=binaryNumber%10;
98          binaryNumber/=10;
99          b6=binaryNumber%10;
100         binaryNumber/=10;
101         b7=0;
102         b8=0;
103
104     }
105
106     //printing the binary equivalent of the secret code.
107     printf("Binary equivalent of the chosen code is ");
108     printf("%d%d%d%d%d%d%d%d",b8,b7,b6,b5,b4,b3,b2,b1);
109     break;
110
111 case 2:
112
113     fflush(stdin);
114     printf("You have chosen option 2!\n");
115     printf("Which base will you use to enter text (base 16/2)? ");
116     scanf("%d",&base);
117     while(base!=2 && base!=16){
118         fflush(stdin);
119         printf("Please enter base either 2 or 16! ");
120         scanf("%d",&base);
121     }
122
123     if(base==2){
124         char ch;
125         int errorFlag=0,errorBinaryFlag=0;//if error flag equals 1, then user need to enter the text again.
126         int asciiValue,lengthCount=0;//length is initialized as -1 as I dont count the enter as a bit.
127
128
129         while(!errorFlag){
```

```
130             printf("Please enter the text to encrypt: ");
131             fflush(stdin);
132             do{
133 
134                 scanf("%c",&ch);
135                 asciiValue=ch;
136 
137 
138                 if(asciiValue!=48&&asciiValue!=49&&asciiValue!=10){//checking whether entered character is
either 0 or 1.
139 
140                     errorBinaryFlag=1;
141                     break;
142 
143                 }
144                 if(asciiValue!=10){// I will not count the enter as a character
145 
146                     lengthCount++;
147 
148                     //applying XOR operation to every bit.
149                     if(lengthCount%8==0){
150                         if(b1!=asciiValue-48){
151 
152                             ecB1=1;
153                         }
154                         else{
155 
156                             ecB1=0;
157                         }
158                         int encodedText;//this is the decimal value of encoded Text.
159                         encodedText=ecB8*pow(2,7)+ecB7*pow(2,6)+ecB6*pow(2,5)+ecB5*pow(2,4)+ecB4*pow(2,3)+
ecB3*pow(2,2)+ecB2*pow(2,1)+ecB1*pow(2,0);
160 
161                         printf("%c",encodedText);//printing the encoded text
162                     }
163                     else if(lengthCount%8==7){
164                         if(b2!=asciiValue-48){
165 
166                             ecB2=1;
167                         }
168                         else{
169 
170                             ecB2=0;
171                         }
172                     }
173                     else if(lengthCount%8==6){
174                         if(b3!=asciiValue-48){
175 
176                             ecB3=1;
177                         }
178                         else{
179                             ecB3=0;
180                         }
181                     }
182                     else if(lengthCount%8==5){
183                         if(b4!=asciiValue-48){
184 
185                             ecB4=1;
186                         }
187                         else{
188                             ecB4=0;
189                         }
190                     }
191                     else if(lengthCount%8==4){
192                         if(b5!=asciiValue-48){
193 
```

```
                        ecB5=1;
                    }
                    else{
                        ecB5=0;
                    }
                }
                else if(lengthCount%8==3){
                    if(b6!=asciiValue-48){

                        ecB6=1;
                    }
                    else{
                        ecB6=0;
                    }
                }
                else if(lengthCount%8==2){
                    if(b7!=asciiValue-48){

                        ecB7=1;
                    }
                    else{
                        ecB7=0;
                    }
                }
                else if(lengthCount%8==1){
                    if(b8!=asciiValue-48){

                        ecB8=1;
                    }
                    else{
                        ecB8=0;
                    }



                }
            }



        }while(ch!=10);

        //these are error conditions, If there is any, the loop will continue.
        if(errorBinaryFlag==1){
                printf("Please enter only digits 0 and 1! ");
                errorBinaryFlag=0;
                lengthCount=0;
                continue;
        }
        else{
            if(lengthCount<8){
                printf("Please enter at least 8 bits for the text! ");
                lengthCount=0;
                continue;
            }
            else{
                errorFlag=1;
            }

        }

    }
```

```
260
261     }
262     else if(base==16){
263                 char ch;
264                 int errorFlag=0,errorHexFlag=0;
265                 int asciiValue,lengthCount=0;
266
267                 printf("Please enter the text to encrypt: ");
268                 while(!errorFlag){
269
270
271                 fflush(stdin);
272                 do{
273
274                     scanf("%c",&ch);
275                     asciiValue=ch;
276                     //the entered character must be between 0-9 or A-F
277                     if((asciiValue<48||asciiValue>57)&&(asciiValue<65||asciiValue>70)&&(asciiValue!=10)){
278
279                         errorHexFlag=1;
280                         break;
281
282                     }
283
284                     if(asciiValue!=10){
285                         lengthCount++;
286                     }
287
288                     //here, I declared 0 or 1 to every bit based on the entered charachter from the user
289                     if(lengthCount%2==1){
290                         switch(ch){
291                             case '0':
292                                 hexToB8=0;
293                                 hexToB7=0;
294                                 hexToB6=0;
295                                 hexToB5=0;
296                                 break;
297                             case '1':
298                                 hexToB8=0;
299                                 hexToB7=0;
300                                 hexToB6=0;
301                                 hexToB5=1;
302                                 break;
303                             case '2':
304                                 hexToB8=0;
305                                 hexToB7=0;
306                                 hexToB6=1;
307                                 hexToB5=0;
308                                 break;
309                             case '3':
310                                 hexToB8=0;
311                                 hexToB7=0;
312                                 hexToB6=1;
313                                 hexToB5=1;
314                                 break;
315                             case '4':
316                                 hexToB8=0;
317                                 hexToB7=1;
318                                 hexToB6=0;
319                                 hexToB5=0;
320                                 break;
321                             case '5':
322                                 hexToB8=0;
323                                 hexToB7=1;
324                                 hexToB6=0;
325                                 hexToB5=1;
```

```
                        break;
                    case '6':
                        hexToB8=0;
                        hexToB7=1;
                        hexToB6=1;
                        hexToB5=0;
                        break;
                    case '7':
                        hexToB8=0;
                        hexToB7=1;
                        hexToB6=1;
                        hexToB5=1;
                        break;
                    case '8':
                        hexToB8=1;
                        hexToB7=0;
                        hexToB6=0;
                        hexToB5=0;
                        break;
                    case '9':
                        hexToB8=1;
                        hexToB7=0;
                        hexToB6=0;
                        hexToB5=1;
                        break;
                    case 'A':
                        hexToB8=1;
                        hexToB7=0;
                        hexToB6=1;
                        hexToB5=0;
                        break;
                    case 'B':
                        hexToB8=1;
                        hexToB7=0;
                        hexToB6=1;
                        hexToB5=1;
                        break;
                    case 'C':
                        hexToB8=1;
                        hexToB7=1;
                        hexToB6=0;
                        hexToB5=0;
                        break;
                    case 'D':
                        hexToB8=1;
                        hexToB7=1;
                        hexToB6=0;
                        hexToB5=1;
                        break;
                    case 'E':
                        hexToB8=1;
                        hexToB7=1;
                        hexToB6=1;
                        hexToB5=0;
                        break;
                    case 'F':
                        hexToB8=1;
                        hexToB7=1;
                        hexToB6=1;
                        hexToB5=1;
                        break;

                }
            }
            else{
                switch(ch){
```

```
                        case '0':
                            hexToB4=0;
                            hexToB3=0;
                            hexToB2=0;
                            hexToB1=0;
                            break;
                        case '1':
                            hexToB4=0;
                            hexToB3=0;
                            hexToB2=0;
                            hexToB1=1;
                            break;
                        case '2':
                            hexToB4=0;
                            hexToB3=0;
                            hexToB2=1;
                            hexToB1=0;
                            break;
                        case '3':

                            hexToB4=0;
                            hexToB3=0;
                            hexToB2=1;
                            hexToB1=1;
                            break;
                        case '4':
                            hexToB4=0;
                            hexToB3=1;
                            hexToB2=0;
                            hexToB1=0;
                            break;
                        case '5':
                            hexToB4=0;
                            hexToB3=1;
                            hexToB2=0;
                            hexToB1=1;
                            break;
                        case '6':
                            hexToB4=0;
                            hexToB3=1;
                            hexToB2=1;
                            hexToB1=0;
                            break;
                        case '7':
                            hexToB4=0;
                            hexToB3=1;
                            hexToB2=1;
                            hexToB1=1;
                            break;
                        case '8':
                            hexToB4=1;
                            hexToB3=0;
                            hexToB2=0;
                            hexToB1=0;
                            break;
                        case '9':
                            hexToB4=1;
                            hexToB3=0;
                            hexToB2=0;
                            hexToB1=1;
                            break;
                        case 'A':
                            hexToB4=1;
                            hexToB3=0;
                            hexToB2=1;
                            hexToB1=0;
```

```
                        break;
                    case 'B':
                        hexToB4=1;
                        hexToB3=0;
                        hexToB2=1;
                        hexToB1=1;
                        break;
                    case 'C':
                        hexToB4=1;
                        hexToB3=1;
                        hexToB2=0;
                        hexToB1=0;
                        break;
                    case 'D':
                        hexToB4=1;
                        hexToB3=1;
                        hexToB2=0;
                        hexToB1=1;
                        break;
                    case 'E':

                        hexToB4=1;
                        hexToB3=1;
                        hexToB2=1;
                        hexToB1=0;
                        break;
                    case 'F':
                        hexToB4=1;
                        hexToB3=1;
                        hexToB2=1;
                        hexToB1=1;
                        break;
                }
            if(ch!=10){


                //If no problem, I will declared encrypted text bits.
                if(b1!=hexToB1){

                    ecB1=1;
                }
                else{

                    ecB1=0;
                }




                if(b2!=hexToB2){

                    ecB2=1;
                }
                else{

                    ecB2=0;
                }


                if(b3!=hexToB3){

                    ecB3=1;
                }
                else{
                    ecB3=0;
```

```
                        }


                        if(b4!=hexToB4){

                            ecB4=1;
                        }
                        else{
                            ecB4=0;
                        }


                        if(b5!=hexToB5){

                            ecB5=1;
                        }
                        else{
                            ecB5=0;
                        }


                        if(b6!=hexToB6){

                            ecB6=1;
                        }
                        else{
                            ecB6=0;
                        }


                        if(b7!=hexToB7){

                            ecB7=1;
                        }
                        else{
                            ecB7=0;
                        }


                        if(b8!=hexToB8){

                            ecB8=1;
                        }
                        else{
                            ecB8=0;
                        }

                        int encodedText;
                        encodedText=ecB8*pow(2,7)+ecB7*pow(2,6)+ecB6*pow(2,5)+ecB5*pow(2,4)+ecB4*pow(2,3)+
ecB3*pow(2,2)+ecB2*pow(2,1)+ecB1*pow(2,0);
                        //printing encoded text
                        printf("%c",encodedText);

                    }


                    }
                }while(ch!=10);


                //these are error conditions, If there is any, the loop will continue.
                if(errorHexFlag==1){
                    printf("Please enter only 0-9 or A-f ");
                    errorHexFlag=0;
                    lengthCount=0;
                    continue;
```

```
                }
                else{
                    if(lengthCount<2){
                        printf("Please enter at least 2 bits for the text! ");
                        lengthCount=0;
                        continue;
                    }
                    else{
                        errorFlag=1;
                    }

                }

        }
    }


    break;

    case 3:
        printf("See You!\n");
        break;


}


}while(op!=3);


return 0;

}
```