```
//WAP to perform to set a bit, clear a bit, toggle a bit, find status of a bit and find flag status of each bit in given value.
/*Algorithm:
1.Input chat to perform particular operation and variable, bit position to perform bitwise operation.
2.using switch case perform particular operation
3.print the desied result in console window.*/
#include <stdio.h>                          //include standard input and output header file using preprocessor directive

//Function Declarations
int extract(int ,int );                          //declare a function to extract a bit to find status of that particular bit.
int setbit(int ,int );                                                    //declare a setbit() to set a particular bit.
int clearbit(int ,int );                                                  //declare a clearbit() to clear a particular bit.
int togglebit(int ,int );                                                 //declare a togglebit() to toggle a particular bit.
int d2b(int );                                                            //declare a d2b() to convert a decimal to binary
void monitor(int ,int ,int );    /*declare a monitor() function to find flag status of  particular bit for sequence of numbers.*/
void testbit(int );              //declare a test() function to find the status of each bit and print the no.of LED's ON/OFF



//main()

int main()
{
        setbuf(stdout,NULL);
   char choice;              //declare a char to input choice
   puts("Enter S-Set a bit || C-Clear a bit || T-Toggle a bit || E-to extract a bit || M-Monitor a bit || t- to test bits(LED's ON/OFF) ");
   scanf("%c",&choice);
   switch(choice)                    //define a switch case to perform different bitwise operations as mentioned above
   {
      case 'S':
      {
      int v_num,v_pos,v_set;                                              //declare variables for decimal and bit position
            puts("enter the number and bit position to set.");
            scanf("%d %d",&v_num,&v_pos);                                 //input decimal and bit position values
            printf("Before setting a bit value is %d & it's binary value is ",v_num);
            d2b(v_num);                                                   //print the binary value of the given number
            v_set=setbit(v_num,v_pos);                    //get the value after setting a particular bit
            printf("\nAfter setting %d-bit of %d,the value is %d it's binary value is ",v_pos,v_num,v_set);
            d2b(v_set);                                                   //print the binary value after bit setting.
            break;
      }
      case 'C':
      {
      int v_num,v_pos,v_clear;                                            //declare variables for decimal and bit position
            puts("enter the number and bit position to clear.");
            scanf("%d %d",&v_num,&v_pos);                                 //input decimal and bit position values
            printf("Before clearing a bit value is %d it's binary value is ",v_num);
            d2b(v_num);                                                   //print the binary value of the given number
            v_clear=clearbit(v_num,v_pos);                  //get the value after clearing a particular bit
            printf("\nAfter clearing %d-bit in %d,the value is %d it's binary value is ",v_pos,v_num,v_clear);
            d2b(v_clear);                                                 //print the binary value after bit clearing.
      break;
      }
      case 'T':
```

```
    {
      int v_num,v_pos,v_togg;                                   //declare variables for decimal and bit position
          puts("enter the number and bit position to toggle.");
          scanf("%d %d",&v_num,&v_pos);                              //input decimal and bit position values
          printf("value before toggling a bit: %d it's binary value is ",v_num);
          d2b(v_num);                                             //print the binary value of the given number
          v_togg=togglebit(v_num,v_pos);                          //get the value after toggling a particular bit
          printf("\nAfter toggling %d-bit in %d,the value is %d it's binary value is ",v_pos,v_num,v_togg);
          d2b(v_togg);                                            //print the binary value after bit toggling.
          break;
    }
    case 'E':
    {
      int v_num,v_pos,v_status;                                 //declare variables for decimal and bit position
            puts("enter the number and bit position to find it's status:");
            scanf("%d %d",&v_num,&v_pos);                            //input decimal and bit position values
            printf("The binary value of %d is : ",v_num);
            d2b(v_num);
            v_status=extract(v_num,v_pos);
            printf("%d-bit is %d ",v_pos,v_status);
            break;
    }
    case 'M':
    {
      int v_min,v_max,v_status;
      printf("Enter the min to max range and the status bit to find flag status:\n");
      scanf("%d %d %d",&v_min,&v_max,&v_status);
      monitor(v_min,v_max,v_status);
      break;
    }
    case 't':
    {
      int v_num;
      printf("Enter the Decimal to find the test each bit and find no of led's ON/OFF:\n");
      scanf("%d",&v_num);
      testbit(v_num);
      break;
    }
    default:
    printf("Invalid Attempt");
  }
}                                                                                   //end main()


//Definitions of functions

int setbit(int n,int p)                                     //define a function to set a particular bit of given value.
{
  int v_set;
  v_set=n|(1<<p);
  return v_set;
}
int clearbit(int n,int p)                                   //define a function to clear a particular bit of given value.
{
  int v_clear;
```

```
   v_clear=n&~(1<<p);
   return v_clear;
}
int togglebit(int n,int p)                                  //define a function to toggle a particular bit of given value.
{
   int v_toggle;
   v_toggle=n^(1<<p);
   return v_toggle;
}
int extract(int n,int p)                                    //define a function to extract a particular bit in a given value.
{
   int v_status;
   v_status=(n>>p)&1;
   return v_status;
}
void monitor(int min,int max,int pos)          //define a function to monitor a particular bit and to find the flag status.
{
   int i_ref,t_num,res;
   for(i_ref=min;i_ref<=max;i_ref++)
   {
      t_num=i_ref;
      res=(t_num>>pos)&1;
      if(res==0)
      {
         printf("For %d flag is not set because %d-bit is  %d\n",i_ref,pos,res);
      }
      else if(res==1)
      {
         printf("For %d flag is set because %d-bit is  %d \n",i_ref,pos,res);
      }
   }
}
int d2b(int v_dec)                                          //define a d2b() function to convert decimal to binary value.
{
int rem[32],coef,i=0,cnt=0;
//declare rem[32] array to store the binary converted decimal, i for referencing, coef and cnt.
          while(v_dec)                                  //define while() loop in order to get the binary value for given decimal
          {
                    rem[i]=v_dec%2;
                    coef=(v_dec/2);
                    v_dec=coef;
                    i++;
                    cnt++;
          }
          if(cnt<8)
          {
             cnt=8;
          }
          for(i=cnt-1;i>=0;i--)                         //define a for loop to print the binary value of given decimal value.
          {
                    if(rem[i]==0||rem[i]==1)
                    {
                    printf("%d ",rem[i]);
                    }
                    else
                    {
```

```
                    rem[i]=0;
                    printf("%d ",rem[i]);
                }
        }
        printf("\n");
        return cnt;
}
void testbit(int v_dec)                         //define a function to extract each bit and to print no.of led's on/off
{
   int ON=0,OFF=0,v_pos,v_tres,max_index;
   max_index=d2b(v_dec);
   for(v_pos=0;v_pos<max_index;v_pos++)
   {
      v_tres=extract(v_dec,v_pos);
      if(v_tres==0)
      {
         OFF++;
      }
      else if(v_tres==1)
      {
         ON++;
      }
   }
   printf("\n%d LED's are High  && %d LED's are Low",ON,OFF);                    //prints no.of led's on/off.
}
```