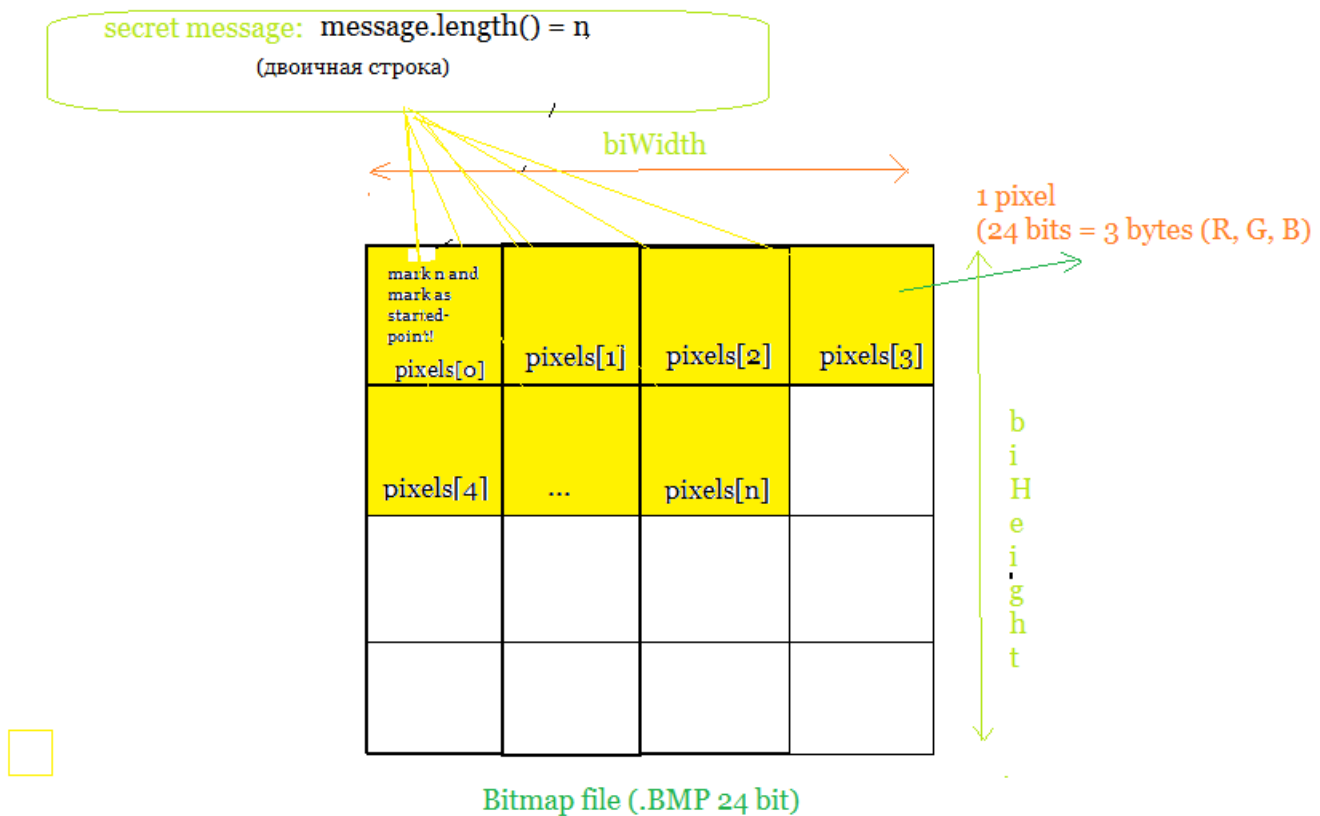


1. Описание алгоритма кодирования.



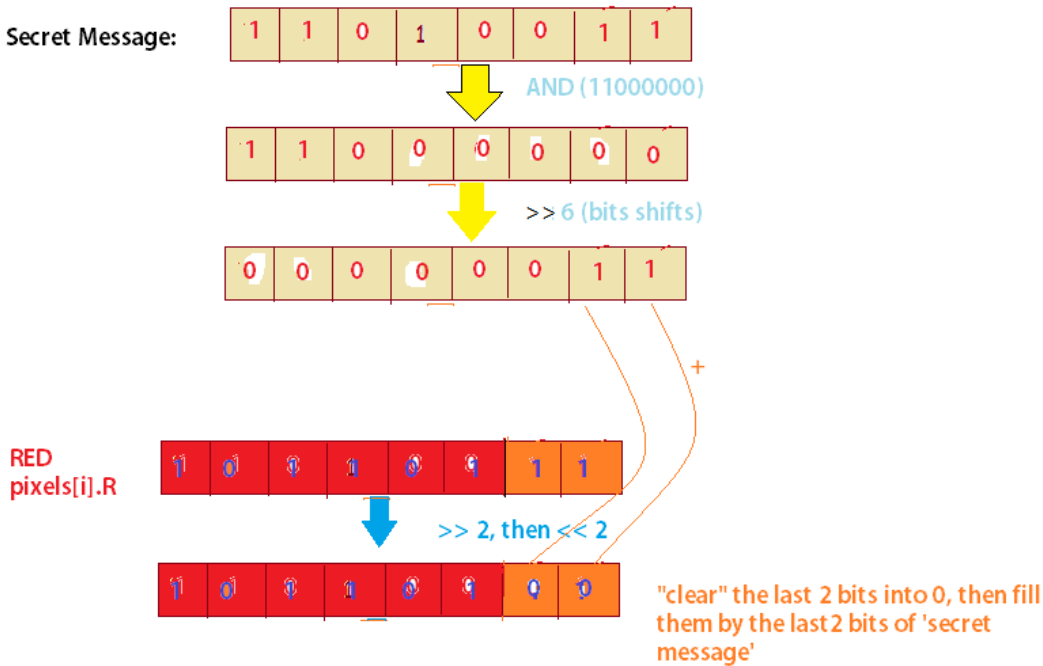
Idea: ☺ *Арифметический сдвиг*

При этом сдвиге слово рассматривается не просто как группа битов, а как целое число в дополнительном коде. При сдвиге влево ведёт себя как логический сдвиг, при сдвиге вправо уходящий бит исчезает, не влияя на оставшиеся биты, а на месте появившегося бита устанавливается бит, соответствующий знаку.

```
for (int i=1;i<=n;i++)
{
    Red=buff[i-1]&192; //11000000          /*BITWISE OPERATION /bits shift
    Red=Red>>6;
    Green=buff[i-1]&56; //00111000
    Green=Green>>3;
    Blue=buff[i-1]&7;
    k=pixels[i].R>>2;
    pixels[i].R=k<<2;
    pixels[i].R=pixels[i].R+Red;
    k=pixels[i].G>>3;
    pixels[i].G=k<<3;
    pixels[i].G=pixels[i].G+Green;
    k=pixels[i].B>>3;
    pixels[i].B=k<<3;
    pixels[i].B=pixels[i].B+Blue;
}
```

Encode: example with RED color.

Secret Message:

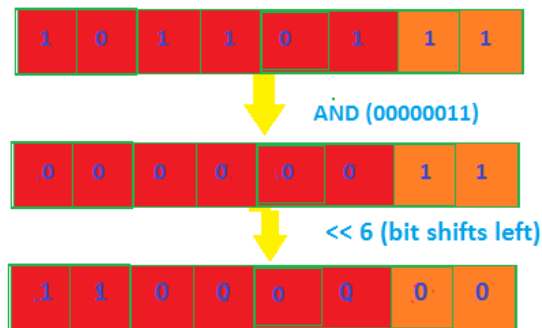


Это примеры для Красного цвета. Цвет Зеленый, Синий, аналогичные.

2. Описание алгоритма декодирования.

Decode: Example with RED color

RED
pixels[i].R



Secret Message

```
for (int i=1;i<=n;i++)
{
    k1=pixels[i].R&3;
    k2=pixels[i].G&7;
    k3=pixels[i].B&7;
    k=(k1<<6);
    k=k+(k2<<3);
    k=k+k3;
    st[i-1]=k;
}
cout<<st;
```