1. Hw1_1

(1) main function:

```
8   int main(){
9       info();
10      if(operation=="encrypt")
11          encrypt();
12      else if(operation=="decrypt")
13          decrypt();
14      return 0;
15  }
```

(2) info():

```
16  void info(){            //input operations and key
17      cout << "Choose operation (encrypt/decrypt): ";
18      cin >> operation;
19      if(operation!="encrypt"&&operation!="decrypt")
20          cout << "Invalid input!";
21      else{
22          cout << "Input the key: ";
23          cin >> key;
24      }
25      return;
26  }
```

(3) encrypt():

```
27  void encrypt(){         //encrypt the plain text
28      string plain_text;
29      int cipher_text;
30      cout << "Input the plain text: ";
31      cin.ignore();       //handle the enter after inputing the key
32      getline(cin, plain_text);
33      cout << "The cipher text is: ";
34      for (int i = 0; i < plain_text.length();i++){
35          if(plain_text[i]>='A'&&plain_text[i]<='Z'){     //handle the letter which is upper-case
36              cipher_text = plain_text[i] + key - 'A';    //rather than start with 0, the upper-case letter start from 'A'(ASCII: 65)
37              cipher_text %= 26;                          //the relative position of cipher_text starting from 0
38              cipher_text += 'A';                         //plus the start position, it will become the absolute position
39              cout << ((char)cipher_text);
40          }
41          else if(plain_text[i]>='a'&&plain_text[i]<='z'){//handle the letter which is lower-case
42              cipher_text = plain_text[i] + key - 'a';    //rather than start with 0, the upper-case letter start from 'a'(ASCII: 97)
43              cipher_text %= 26;                          //the relative position of cipher_text starting from 0
44              cipher_text += 'a';                         //plus the start position, it will become the absolute position
45              cout << ((char)cipher_text);
46          }
47          else
48              cout << plain_text[i];      //if it's not a english letter, output it directly
49      }
50      cout << endl;
51      return;
52  }
```

(4) decrypt()

```
53  void decrypt(){                         //decryption is roughly in contrast to encryption
54      string cipher_text;
55      int plain_text;
56      cout << "Input the cipher text: ";
57      cin.ignore();
58      getline(cin, cipher_text);
59      cout << "The plain text is: ";
60      for (int i = 0; i < cipher_text.length();i++){
61          if(cipher_text[i]>='A'&&cipher_text[i]<='Z'){       //handle the letter which is upper-case
62              plain_text = 'Z' - (cipher_text[i] - key);
63              plain_text %= 26;
64              plain_text = 'Z' - plain_text;
65              cout << ((char)plain_text);
66          }
67          else if(cipher_text[i]>='a'&&cipher_text[i]<='z'){  //handle the letter which is lower-case
68              plain_text = 'z' - (cipher_text[i] - key);
69              plain_text %= 26;
70              plain_text = 'z' - plain_text;
71              cout << ((char)plain_text);
72          }
73          else
74              cout << cipher_text[i];         //if it's not a english letter, output it directly
75      }
76      cout << endl;
77      return;
78  }
```

(5) result

```
Choose operation (encrypt/decrypt): encrypt
Input the key: 3
Input the plain text: There is a zebra
The cipher text is: Wkhuh lv d cheud
```

```
Choose operation (encrypt/decrypt): decrypt
Input the key: 5
Input the cipher text: N qnpj fuuqjx
The plain text is: I like apples
```

2. Hw1_2

(1) main function()

```
7   int main(){
8       input();
9       brute_force();
10      return 0;
11  }
```

(2) input()

```
12  void input(){       //let user input the cipher text
13      cout << "Input the cipher text: ";
14      getline(cin, cipher_text);
15      return;
16  }
```

(3) brute_force()

```
void brute_force(){     //decrypt the cipher_text based on keys from 1 to 25
    for (int key = 1; key <= 25;key++){
        cout << "key = " << key << ": ";
        for (int i = 0; i < cipher_text.length();i++){
            if (cipher_text[i] >= 'A' && cipher_text[i] <= 'Z'){    //handle the letter which is upper-case
                plain_text = 'Z' - (cipher_text[i] - key);
                plain_text %= 26;
                plain_text = 'Z' - plain_text;
                cout << ((char)plain_text);
            }
            else if (cipher_text[i] >= 'a' && cipher_text[i] <= 'z'){//handle the letter which is lower-case
                plain_text = 'z' - (cipher_text[i] - key);
                plain_text %= 26;
                plain_text = 'z' - plain_text;
                cout << ((char)plain_text);
            }
            else
                cout << cipher_text[i];
        }
        cout << endl;
    }
}
```

(4) result

```
Input the cipher text: Rfc ucyrfcp gq fmr
key = 1: Qeb tbxqebo fp elq
key = 2: Pda sawpdan eo dkp
key = 3: Ocz rzvoczm dn cjo
key = 4: Nby qyunbyl cm bin
key = 5: Max pxtmaxk bl ahm
key = 6: Lzw owslzwj ak zgl
key = 7: Kyv nvrkyvi zj yfk
key = 8: Jxu muqjxuh yi xej
key = 9: Iwt ltpiwtg xh wdi
key = 10: Hvs ksohvsf wg vch
key = 11: Gur jrngure vf ubg
key = 12: Ftq iqmftqd ue taf
key = 13: Esp hplespc td sze
key = 14: Dro gokdrob sc ryd
key = 15: Cqn fnjcqna rb qxc
key = 16: Bpm emibpmz qa pwb
key = 17: Aol dlhaoly pz ova
key = 18: Znk ckgznkx oy nuz
key = 19: Ymj bjfymjw nx mty
key = 20: Xli aiexliv mw lsx
key = 21: Wkh zhdwkhu lv krw
key = 22: Vjg ygcvjgt ku jqv
key = 23: Uif xfbuifs jt ipu
key = 24: The weather is hot
key = 25: Sgd vdzsgdq hr gns
```