

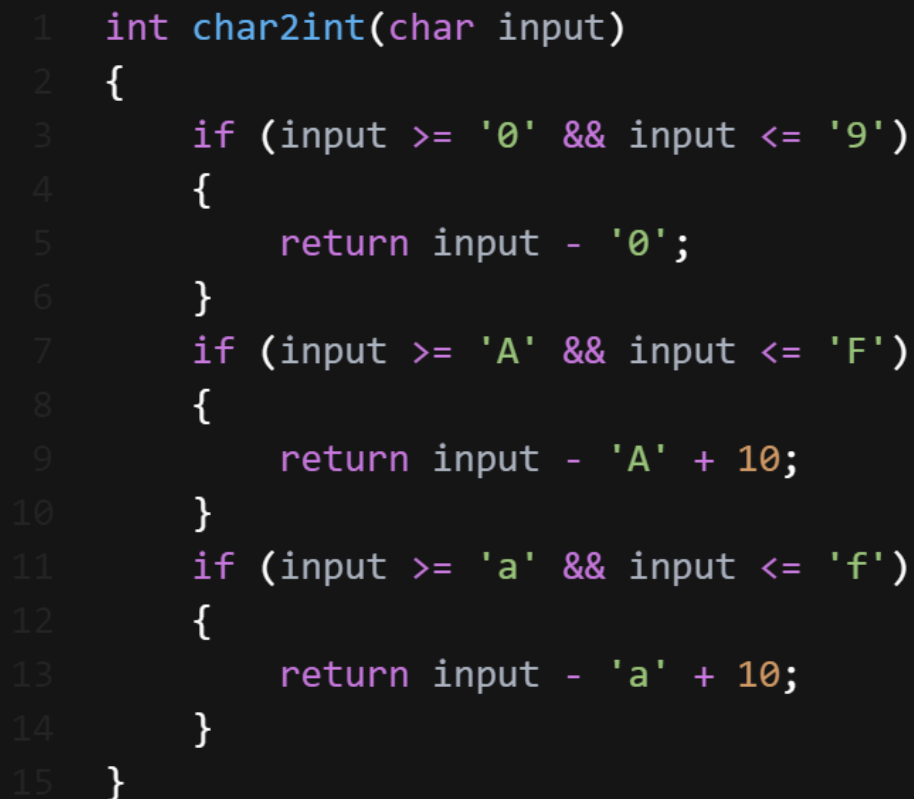
Program decrypt

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int char2int(char input)
5  {
6      if (input >= '0' && input <= '9')
7      {
8          return input - '0';
9      }
10     if (input >= 'A' && input <= 'F')
11     {
12         return input - 'A' + 10;
13     }
14     if (input >= 'a' && input <= 'f')
15     {
16         return input - 'a' + 10;
17     }
18 }
19
20 int main()
21 {
22
23     char KEY1[] = "a6c8b6733c9b22de7bc0253266a3867df55acde8635e19c73313";
24     char KEY2_KEY3[] = "c1545756687e7573db23aa1c3452a098b71a7fbf0fddddd5fc1";
25     char FLAG_KEY1_KEY2_KEY3[] = "04ee9855208a2cd59091d04767ae47963170d1660df7f56f5faf";
26     char FLAG_KEY1[] = {0};
27     char FLAG[] = {0};
28     int lenght = strlen(KEY1);
29
30     for (size_t a = 0; a < lenght; a++)
31     {
32         KEY1[a] = char2int(KEY1[a]);
33     }
34     for (size_t i = 0; i < lenght; i++)
35     {
36         KEY2_KEY3[i] = char2int(KEY2_KEY3[i]);
37     }
38     for (size_t j = 0; j < lenght; j++)
39     {
40         FLAG_KEY1_KEY2_KEY3[j] = char2int(FLAG_KEY1_KEY2_KEY3[j]);
41     }
42     for (size_t k = 0; k < lenght; k++)
43     {
44         FLAG_KEY1[k] = KEY2_KEY3[k] ^ FLAG_KEY1_KEY2_KEY3[k];
45         FLAG_KEY1[k] = char2int(FLAG_KEY1[k]);
46     }
47     for (size_t l = 0; l < lenght; l++)
48     {
49         FLAG[l] = FLAG_KEY1[l] ^ KEY1[l];
50     }
51
52     for (size_t i = 0; i < lenght; i = i+2)
53     {
54         char total = (FLAG[i]*16) + FLAG[i+1];
55         printf("%c",total);
56     }
57
58     return 0;
59 }
```

- Methods ແລະ Library ທີ່ໃຊ້

<string.h> ໃຊ້ເພື່ອຫາຄວາມຍາວຂອງ string.

- Function char2int() ເປັນ function ທີ່ໃຊ້ໃນການປ່ຽນຊະນິດຂໍ້ມູນໂຕອັກສອນໃຫ້ມາຢູ່ໃນຮູບແບບຂອງ integer.



```
1  int char2int(char input)
2  {
3      if (input >= '0' && input <= '9')
4      {
5          return input - '0';
6      }
7      if (input >= 'A' && input <= 'F')
8      {
9          return input - 'A' + 10;
10     }
11     if (input >= 'a' && input <= 'f')
12     {
13         return input - 'a' + 10;
14     }
15 }
```

- Logic ທີ່ໃຊ້

ຈາກບົດໂຈດໃຫ້ key ມາ 3 ອັນ ໂດຍແຕ່ລະ key ມີຮູບແບບຄືກັບ hexadecimal ເລີຍ. ເມື່ອຮູ້ວ່າເປັນ hex ແລ້ວ ນ້ອງກໍຈະໃຊ້ວິທີ loop key ເພື່ອໃຫ້ໄດ້ຕົວອັກສອນແຍກກັນ ແລ້ວແຕ່ລະຕົວອັກສອນນັ້ນນ້ອງກໍຈະແປງໃຫ້ມັນເປັນ integer ເພື່ອທີ່ເຮົາຈະສາມາດນຳໃຊ້ XOR ໄດ້.

ໂດຍນ້ອງຈະເອົາ FLAG_KEY1_KEY2_KEY3 ມາ XOR ໃຫ້ແກ່ KEY2_KEY3 ແລ້ວເຮົາຈະໄດ້ FLAG_KEY1 ມາ.

```
1  for (size_t a = 0; a < lenght; a++)
2  {
3      KEY1[a] = char2int(KEY1[a]);
4  }
5  for (size_t i = 0; i < lenght; i++)
6  {
7      KEY2_KEY3[i] = char2int(KEY2_KEY3[i]);
8  }
9  for (size_t j = 0; j < lenght; j++)
10 {
11     FLAG_KEY1_KEY2_KEY3[j] = char2int(FLAG_KEY1_KEY2_KEY3[j]);
12 }
```

ເມື່ອໄດ້ FLAG_KEY1 ກໍຈະ loop ເອົາຄ່າແຕ່ລະໂຕແລ້ວໃຊ້ char2int() ເພື່ອແປງເປັນ integer ແລ້ວຈຶ່ງນຳມາ XOR ໃຫ້ແກ່ KEY1 ເຮົາກໍຈະໄດ້ FLAG ທີ່ມີຊະນິດຂໍ້ມູນເປັນ integer ມາ.

ເພື່ອຈັດໃຫ້ມັນເປັນ hex 2digits. ໂດຍ FLAG ໄດ້ມາ ນ້ອງຈະເອົາມາ loop ໂດຍຈະ loop ເພີ່ມຂຶ້ນເທື່ອລະ 2. ແລ້ວນຳເອົາຄ່າເລີ່ມຕົ້ນ (i) ແປງເປັນເລກຖານ 16 ດ້ວຍການເອົາ (i) ຄູນ 16 ແລ້ວບວກໃຫ້ຄ່າຖັດໄປ (i+1) ກໍຈະໄດ້ເປັນ hex 2digits.

```
1  for (size_t i = 0; i < lenght; i = i+2)
2  {
3      char total = (FLAG[i]*16) + FLAG[i+1];
4      printf("%c",total);
5  }
```

ຫຼັງຈາກໄດ້ hex 2digits ແລ້ວ ກໍຈະໃຫ້ print ອອກມາເປັນ char ເລີຍ.

ຕົວຢ່າງ ຖ້າຄ່າ ເລີ່ມຕົ້ນເປັນ 63 ຈະເອົາ $(6*16) + 3 = 99$ ເຊິ່ງຖ້າເຮົາ print 99 ດ້ວຍ %c ຈະໄດ້ c ອອກມາ