

## 정보검색과 데이터마이닝 20163170 최은주 KS 완성형, 유니코드 실습

1. KS 완성형(KS X 1001)한글 코드 표를 출력하시오.

1) [A1~FE][A1~FE] 영역 전체 코드 표를 출력하시오.

The screenshot shows a C program named `ir_lab1.c` with the following code:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void ksx1001_table()
5  {
6      int i, j;
7      int hcode;
8      int cnt = 0;
9      for (i = 0xA1; i <= 0xFE; i++) {
10         for (j = 0xA1; j <= 0xFE; j++) {
11             hcode = (i << 8) | j;
12             printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
13             cnt++;
14         }
15     }
16     printf("cnt : %d", cnt);
17 }
18
19 int main()
20 {
21     ksx1001_table();
22     system("pause");
23     return 0;
24 }
25
```

The output on the right shows the first 20 entries of the table, ranging from 0xfee6 to 0xfefe, with a total count of 8836.

```
: 0xfee6 <65254>
: 0xfee7 <65255>
: 0xfee8 <65256>
: 0xfee9 <65257>
: 0xfeea <65258>
: 0xfeeb <65259>
: 0xfeec <65260>
: 0xfeed <65261>
: 0xfeee <65262>
: 0xfeef <65263>
: 0xfef0 <65264>
: 0xfef1 <65265>
: 0xfef2 <65266>
: 0xfef3 <65267>
: 0xfef4 <65268>
: 0xfef5 <65269>
: 0xfef6 <65270>
: 0xfef7 <65271>
: 0xfef8 <65272>
: 0xfef9 <65273>
: 0xfefa <65274>
: 0xfefb <65275>
: 0xfefc <65276>
: 0xfefd <65277>
: 0xfefe <65278>
cnt : 8836계속하려면
```

2) 한글 2,350자(B0~C8)만 출력하시오.

The screenshot shows a C program named `ir_lab1.c` with the following code:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void ksx1001_table()
5  {
6      int i, j;
7      int hcode;
8      int cnt = 0;
9      for (i = 0xB0; i <= 0xC8; i++) {
10         for (j = 0xA1; j <= 0xFE; j++) {
11             hcode = (i << 8) | j;
12             printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
13             cnt++;
14         }
15     }
16     printf("cnt : %d", cnt);
17 }
18
19 int main()
20 {
21     ksx1001_table();
22     system("pause");
23     return 0;
24 }
25
```

The output on the right shows the first 20 entries of the table, ranging from 0xc8e2 to 0xc8fe, with a total count of 2350.

```
: 0xc8e2 <51426>
: 0xc8e3 <51427>
: 0xc8e4 <51428>
: 0xc8e5 <51429>
: 0xc8e6 <51430>
: 0xc8e7 <51431>
: 0xc8e8 <51432>
: 0xc8e9 <51433>
: 0xc8ea <51434>
: 0xc8eb <51435>
: 0xc8ec <51436>
: 0xc8ed <51437>
: 0xc8ee <51438>
: 0xc8ef <51439>
: 0xc8f0 <51440>
: 0xc8f1 <51441>
: 0xc8f2 <51442>
: 0xc8f3 <51443>
: 0xc8f4 <51444>
: 0xc8f5 <51445>
: 0xc8f6 <51446>
: 0xc8f7 <51447>
: 0xc8f8 <51448>
: 0xc8f9 <51449>
: 0xc8fa <51450>
: 0xc8fb <51451>
: 0xc8fc <51452>
: 0xc8fd <51453>
: 0xc8fe <51454>
cnt : 2350계속하려면 0
```

3) 한자 4,888자(CA~FD)만 출력하시오.

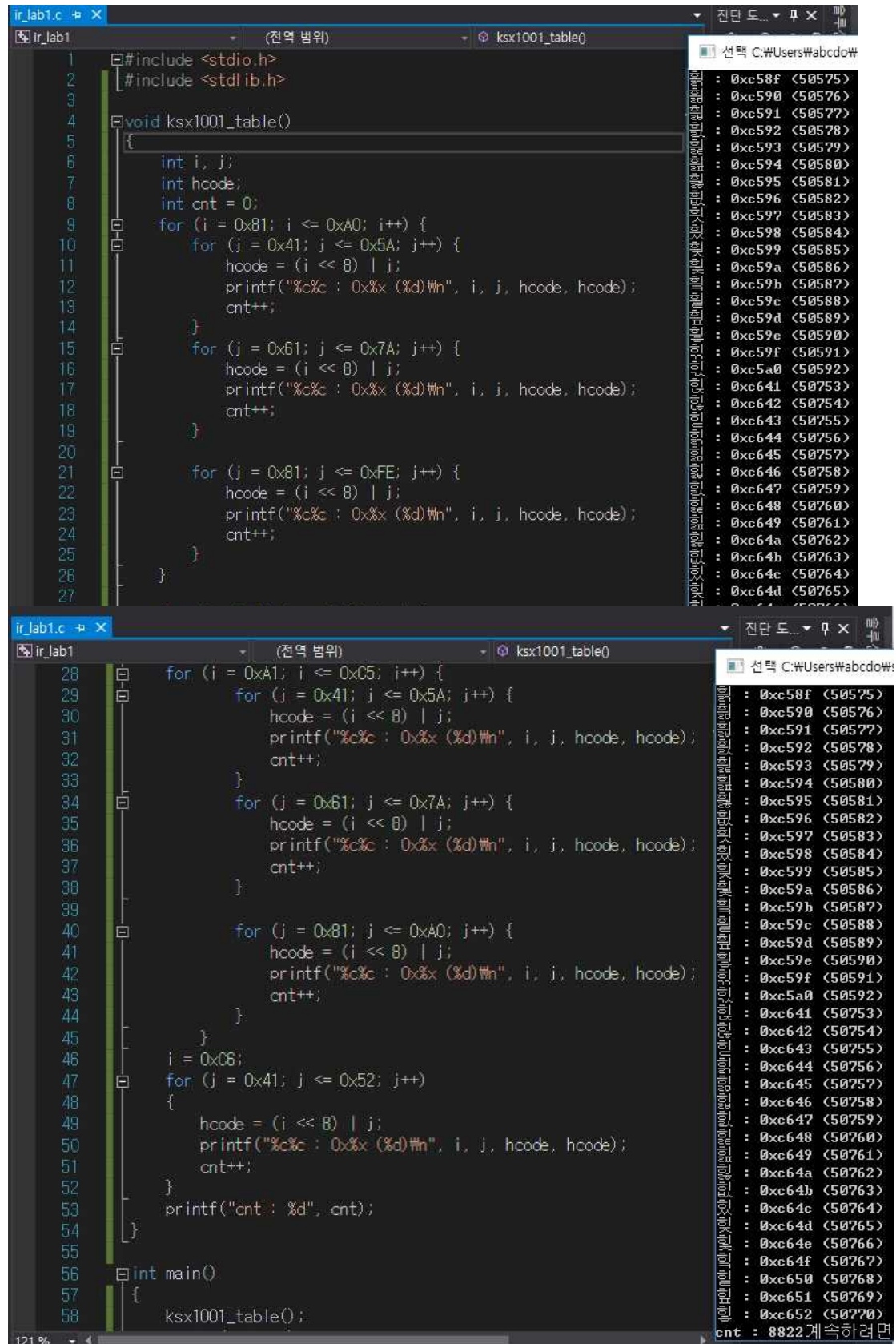
```
ir_lab1.c  ir_lab1  (전역 범위)  ksx1001_table()  진단 도구  C:\Users\abdcdo\source\repos

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void ksx1001_table()
5  {
6      int i, j;
7      int hcode;
8      int cnt = 0;
9      for (i = 0xCA; i <= 0xFD; i++) {
10         for (j = 0xA1; j <= 0xFE; j++) {
11             hcode = (i << 8) | j;
12             printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
13             cnt++;
14         }
15     }
16     printf("cnt : %d", cnt);
17 }
18
19 int main()
20 {
21     ksx1001_table();
22     system("pause");
23     return 0;
24 }
25
```

欠 : 0xfde2 <64994>  
歎 : 0xfde3 <64995>  
歎 : 0xfde4 <64996>  
歎 : 0xfde5 <64997>  
歎 : 0xfde6 <64998>  
歎 : 0xfde7 <64999>  
歎 : 0xfde8 <65000>  
歎 : 0xfde9 <65001>  
歎 : 0xfdea <65002>  
歎 : 0xfdeb <65003>  
歎 : 0xfdec <65004>  
歎 : 0xfded <65005>  
歎 : 0xfdee <65006>  
歎 : 0xfdef <65007>  
歎 : 0xfdf0 <65008>  
歎 : 0xfdf1 <65009>  
歎 : 0xfdf2 <65010>  
歎 : 0xfdf3 <65011>  
歎 : 0xfdf4 <65012>  
歎 : 0xfdf5 <65013>  
歎 : 0xfdf6 <65014>  
歎 : 0xfdf7 <65015>  
歎 : 0xfdf8 <65016>  
歎 : 0xfdf9 <65017>  
歎 : 0xfdfa <65018>  
歎 : 0xfdfb <65019>  
歎 : 0xfdfc <65020>  
歎 : 0xfdfd <65021>  
歎 : 0xfdfе <65022>  
cnt : 4888 계속하려면 아무

#### 4) CP949의 8,822자를 출력하시오.

##### 4-1 사진



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

void ksx1001_table()
{
    int i, j;
    int hcode;
    int cnt = 0;
    for (i = 0x81; i <= 0xA0; i++) {
        for (j = 0x41; j <= 0x5A; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
        for (j = 0x61; j <= 0x7A; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
        for (j = 0x81; j <= 0xFE; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
    }

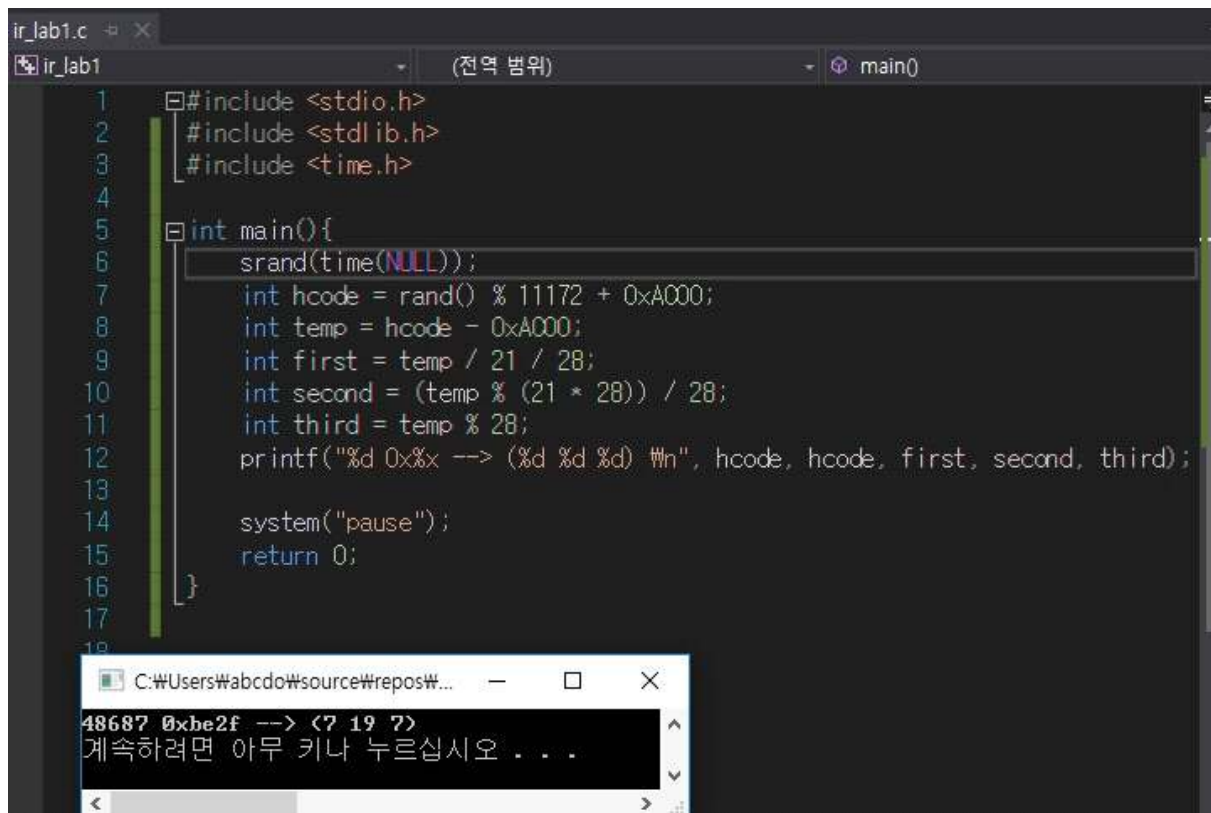
    for (i = 0xA1; i <= 0xC5; i++) {
        for (j = 0x41; j <= 0x5A; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
        for (j = 0x61; j <= 0x7A; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
        for (j = 0x81; j <= 0xA0; j++) {
            hcode = (i << 8) | j;
            printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
            cnt++;
        }
    }
    i = 0xC6;
    for (j = 0x41; j <= 0x52; j++) {
        hcode = (i << 8) | j;
        printf("%c%c : 0x%x (%d)\n", i, j, hcode, hcode);
        cnt++;
    }
    printf("cnt : %d", cnt);
}

int main()
{
    ksx1001_table();
}
```

Output:

```
0xc58f (50575)
0xc590 (50576)
0xc591 (50577)
0xc592 (50578)
0xc593 (50579)
0xc594 (50580)
0xc595 (50581)
0xc596 (50582)
0xc597 (50583)
0xc598 (50584)
0xc599 (50585)
0xc59a (50586)
0xc59b (50587)
0xc59c (50588)
0xc59d (50589)
0xc59e (50590)
0xc59f (50591)
0xc5a0 (50592)
0xc641 (50753)
0xc642 (50754)
0xc643 (50755)
0xc644 (50756)
0xc645 (50757)
0xc646 (50758)
0xc647 (50759)
0xc648 (50760)
0xc649 (50761)
0xc64a (50762)
0xc64b (50763)
0xc64c (50764)
0xc64d (50765)
0xc64e (50766)
0xc64f (50767)
0xc650 (50768)
0xc651 (50769)
0xc652 (50770)
cnt : 8822 계속하려면
```

2. 임의의 유니코드 값에 대한 한글 코드 값(10진수 또는 16진수)과 <초성, 중성, 종성>값을 출력하시오.



The image shows a C program in a code editor and its execution output in a console window. The code generates a random Korean code point and decomposes it into its constituent parts: the full code point, the initial consonant (choseong), the medial consonant (jungseong), and the final consonant (jongseong).


```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main(){
6      srand(time(NULL));
7      int hcode = rand() % 11172 + 0xA000;
8      int temp = hcode - 0xA000;
9      int first = temp / 21 / 28;
10     int second = (temp % (21 * 28)) / 28;
11     int third = temp % 28;
12     printf("%d 0x%x --> (%d %d %d) %n", hcode, hcode, first, second, third);
13
14     system("pause");
15     return 0;
16 }
17
18
```

The console output shows the result of the program's execution:

```
C:\Users\Wabdo\source\repos\... 48687 0xbe2f --> <7 19 7>
계속하려면 아무 키나 누르십시오 . . .
```



### 3. 임의의 유니코드 음절을 자모를 분리하며 출력하시오.



```
#include <stdio.h> #include <stdlib.h> #include <time.h>

int main(){
    srand(time(NULL));
    int hcode = rand() % 11172 + 0xAC00;
    int temp = hcode - 0xAC00;
    int first = temp / 21 / 28;
    int second = (temp % (21 * 28)) / 28;
    int third = temp % 28;

    int a = first + 0x1100;
    int a1 = a >> 8;
    int a2 = a & 0x00FF;

    int b = second + 0x1161;
    int b1 = b >> 8;
    int b2 = b & 0x00FF;

    int c = third + 0x11A7;
    int c1 = c >> 8;
    int c2 = c & 0x00FF;

    int i = hcode >> 8;
    int j = hcode & 0x00FF;

    putchar(0xFE);
    putchar(0xFF);

    putchar(i);
    putchar(j);

    putchar(a1);
    putchar(a2);

    putchar(b1);
    putchar(b2);

    putchar(c1);
    putchar(c2);

    //초성: U+1100~U+1112, 중성: U+1161~U+1175, 종성: U+11A8~U+11C2
    return 0;
}
```

```
C:\Users\Wabcd\source\repos\ir_lab1>gcc ir_lab1.c
C:\Users\Wabcd\source\repos\ir_lab1>a.exe > output.txt
```

```
output - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
라 = ㅏ
```

처음에는 파일입출력을 사용하여 3번문제를 실습해보려고 했지만

output.txt에 초성중성종성이 제대로 나오지않아

교수님께서 오늘 수업시간에 알려주신 putchar를 사용하여 실습해보았습니다.