# 빅데이터 HW1 (한글 음절빈도 계산)

컴퓨터공학부

20163084 권보경

1) 한글 test.txt 가 KS 완성형인지, UTF8 인지 판단하는 함수

```
int KS_or_UTF8(FILE* f)
{
	char c1, c2;
	c1 = fgetc(f);

	if((c1 & 0xf0) == 0xe0) //utf8
	{
		printf("utf8₩n");
		return 1;
	}
	else //ks
	{
		printf("ks₩n");
		return 0;
	}
}
```

2) 음절빈도 계산 함수를 2 가지로 구현 KS 완성형 텍스트인 경우 UTF8 텍스트인 경우

* KS 완성형

```
void freqKS(FILE* f)
{
	int c1, c2;

	f = fopen("test.txt", "r");

	while(!feof(f))
	{
		c1 = fgetc(f);
		if(feof(f)) break;
```

```
                if ((c1 & 0x80) == 0)
                {       // MSB == 0
                        countKS[0][c1]++;
                        //      printf("freq[%c%c]      =       %d₩n",      0,      c1,
countKS[0][c1]);

                } else {
                        c2 = fgetc(f);
                        countKS[c1][c2]++; // DBCS
                        //printf("freq[%d%d]          =          %d₩n",          c1,
c2,countKS[c1][c2]);

                }
        }
}
```

* UTF8

```
void freqUTF(FILE* f)
{
        int c1, c2, c3;

        f = fopen("test.txt", "r");

        while(!feof(f))
        {
            c1 = fgetc(f);
                 if(feof(f)) break;

            if ((c1 & 0x80) == 0) {     // MSB == 0
                countUTF[c1]++;
                                // printf("freq[] = %d₩n", countUTF[c1]);

            } else {
                 c2 = fgetc(f);
                 c3 = fgetc(f);
                 int i = (c1 & 0x0f) << 12 | (c2 & 0x3f) << 6 | (c3 & 0x3f);
                 countUTF[i]++;        // DBCS
                                 // printf("freq[] = %d₩n", countUTF[i]);
            }
        }
}
```

3) 빈도수 출력 (KS 완성형, UTF8 각각에 대해)

* KS 완성형

```
void output_KSC5601()
{
    int i, j;

    for (i=0xA1; i <= 0xFE; i++) {
        for (j=0xA1; j <= 0xFE; j++) {
            if (countKS[i][j] >= 1)
                printf("freq[%c%c] = %d\n", i, j, countKS[i][j]);
        }
    }
}
```

* UTF8

```
void output_UTF8()
{
    int i,j;
        char utf8[4] = {0};

    for (i=0xAC00, j=0; j<11172; i++, j++) {
        utf8[0] = 0xE0; utf8[0] |= ((i>>12) &0x000F);
        utf8[1] = 0x80; utf8[1] |= ((i>>6) &0x003F);
        utf8[2] = 0x80; utf8[2] |= (i&0x003F);

       if (countUTF[i] >= 1)
        printf("freq[%c%c%c]      =      %d₩n",      utf8[0],utf8[1],utf8[2],
countUTF[i]);
    }
}
```

* test.txt

권보경컴퓨터권보경경

* KS 완성형 출력결과

```
gwonbogyeong-ui-MacBook-Pro:鏽 ﷺ 댄 ﷺ bokyeong$ ./a.out
ks
freq[경 ] = 3
freq[권 ] = 2
freq[보 ] = 2
freq[컴 ] = 1
freq[터 ] = 1
freq[퓨 ] = 1
```

* UTF8 출력 결과

```
gwonbogyeong-ui-MacBook-Pro:빅 데 이 터  bokyeong$ ./a.out
utf8
freq[경 ] = 3
freq[권 ] = 2
freq[보 ] = 2
freq[컴 ] = 1
freq[터 ] = 1
freq[퓨 ] = 1
```

* 전체 코드

```
#include <stdio.h>

int countKS[256][256]={0,};
int countUTF[65536];

int KS_or_UTF8(FILE* f)
{
        char c1, c2;
        c1 = fgetc(f);

        if((c1 & 0xf0) == 0xe0) //utf8
        {
                printf("utf8\n");
                return 1;
        }
        else //ks
        {
                printf("ks\n");
```

```
                return 0;
        }
}

void freqKS(FILE* f)
{
        int c1, c2;

        f = fopen("test.txt", "r");

        while(!feof(f))
        {
                c1 = fgetc(f);
                if(feof(f)) break;

                if ((c1 & 0x80) == 0)
                {       // MSB == 0
                        countKS[0][c1]++;   // ◆ъ◆Ű◆Ǔ◆
                        // printf("freq[%c%c] = %d\n", 0, c1, countKS[0][c1]);

                } else {
                        c2 = fgetc(f);
                        countKS[c1][c2]++; // DBCS
                        //      printf("freq[%d%d]      =       %d\n",      c1,
c2,countKS[c1][c2]);

                }
        }
}

void output_KSC5601()
{
    int i, j;

    for (i=0xA1; i <= 0xFE; i++) {
          for (j=0xA1; j <= 0xFE; j++) {
                                        if (countKS[i][j] >= 1)
                                            printf("freq[%c%c]   =   %d\n",   i,   j,
countKS[i][j]);
           }
    }
}
```

```
void freqUTF(FILE* f)
{
        int c1, c2, c3;

        f = fopen("test.txt", "r");

        while(!feof(f))
        {
            c1 = fgetc(f);
                 if(feof(f)) break;

            if ((c1 & 0x80) == 0) {     // MSB == 0
                countUTF[c1]++;       // ◆ѕ◆ŰЈ◆
                                  // printf("freq[ ] = %d\n", countUTF[c1]);

            } else {
                 c2 = fgetc(f);
                 c3 = fgetc(f);
                 int i = (c1 & 0x0f) << 12 | (c2 & 0x3f) << 6 | (c3 & 0x3f);
                 countUTF[i]++;        // DBCS
                                   // printf("freq[ ] = %d\n", countUTF[i]);

            }
        }
}

void output_UTF8()
{
    int i,j;
          char utf8[4] = {0};

    for (i=0xAC00, j=0; j<11172; i++, j++) {
                 utf8[0] = 0xE0; utf8[0] |= ((i>>12) &0x000F);
                 utf8[1] = 0x80; utf8[1] |= ((i>>6) &0x003F);
                 utf8[2] = 0x80; utf8[2] |= (i&0x003F);

          if (countUTF[i] >= 1)
                                        printf("freq[%c%c%c]        =        %d\n",
utf8[0],utf8[1],utf8[2], countUTF[i]);
    }
}
```

```
int main()
{
	char c1, c2, c3;

	// char fname(e, "r");
	FILE *f = fopen("test.txt", "r");

	if(KS_or_UTF8(f) == 1)
	{
		freqUTF(f);
		output_UTF8();
	}
	else
	{
		freqKS(f);
		output_KSC5601();
	}


	return 0;
}
```