

The function has a bug, because in

```
const tchar *se = s + (len * sizeof(tchar)); :
```

the multiply with sizeof(tchar) is nonsense, pointer arithmetic is defined to work with object size already;

And the function makes too many assumptions too, namely it accesses memory after len, which may work for String objects, but not in general case:

A valid call may be

```
char* s= new char('A');  
string.FindFirstOf(1, s, 0);
```

So I would suggest to change this function to

```
int AString<B>::FindFirstOf(int len, const tchar *s, int from) const  
{  
    ASSERT(from >= 0 && from <= GetLength());  
    const tchar *ptr = B::Begin();  
    const tchar *e = End();  
    const tchar *se = s + len;  
    if(len == 1) {  
        tchar c1 = s[0];  
        for(const tchar *bs = ptr + from; bs < e; bs++) {  
            if(*bs == c1)  
                return (int)(bs - ptr);  
        }  
        return -1;  
    }  
    if(len == 2) {  
        tchar c1 = s[0];  
        tchar c2 = s[1];  
        for(const tchar *bs = ptr + from; bs < e; bs++) {  
            tchar ch = *bs;  
            if(ch == c1 || ch == c2)  
                return (int)(bs - ptr);  
        }  
        return -1;  
    }  
}
```

```

if(len == 3) {
    tchar c1 = s[0];
    tchar c2 = s[1];
    tchar c3 = s[2];
    for(const tchar *bs = ptr + from; bs < e; bs++) {
        tchar ch = *bs;
        if(ch == c1 || ch == c2 || ch == c3)
            return (int)(bs - ptr);
    }
    return -1;
}
if(len == 4) {
    tchar c1 = s[0];
    tchar c2 = s[1];
    tchar c3 = s[2];
    tchar c4 = s[3];
    for(const tchar *bs = ptr + from; bs < e; bs++) {
        tchar ch = *bs;
        if(ch == c1 || ch == c2 || ch == c3 || ch == c4)
            return (int)(bs - ptr);
    }
    return -1;
}
for(const tchar *bs = ptr + from; bs < e; bs++)
    for(const tchar *ss = s; ss < se; ss++)
        if(*bs == *ss)
            return (int)(bs - ptr);
return -1;
}

```

Regards,
Hans
