

---

Subject: operation with ascii table character (Pascal x C++/U++)

Posted by [BetoValle](#) on Fri, 16 Apr 2021 22:11:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I'm trying to convert a routine from Pascal to U ++, which involves an operation with the number of the character in the ascii table. The problem is that in Pascal

a = ord (result [i]) = 97 , b = ord (result [i]) = 101

a xor b results 4 and this result in U ++ (a^b) is not considered for screen printing or string accumulation: char(4)

in Pascal, char (4) is printed "x04"!

Do you have an equivalent routine?

How to solve? is there any way?

Thanks

---

---

Subject: Re: operation with ascii table character (Pascal x C++/U++)

Posted by [Oblivion](#) on Sat, 17 Apr 2021 08:23:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hello BetoValle,

```
for(int i = 32; i < 128; ++i) {
    Cout() << "\t0x" << FormatIntHex(i, 2)
        << "\t0x" << FormatIntHex(i + 1, 2)
        << "\t0x" << FormatIntHex(i ^ (i + 1), 2) << "\n";
}
```

// OR

```
Cout() << "\t-----\n";
```

```
for(int i = 32; i < 128; ++i) {
    Cout() << Format("\t0x%02x\t0x%02x\t0x%02x\n", i, i + 1, i ^ (i + 1));;
}
```

Something like this?

You can use text formatting and/or formatters, as you can see in the above example.

Best regards,



In pascal I can accumulate the characters 1 to 32 in the string using a loop  
String [i] = chr (i)  
but in U ++ I can't or if it exists then I haven't learned it yet.

---

Subject: Re: operation with ascii table character (Pascal x C++/U++)  
Posted by [Oblivion](#) on Sat, 17 Apr 2021 16:20:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello BetoValle,

I see. AFAIK there is no such formatter in U++ (it is too specific, IMO).

But it is easy to create one, using operator overloading, for example.

```
String& operator*(String& s, int c)
{
    if(c >= 0x20 && c <= 0x7E)
        s.Cat(c);
    else
        s << "0x" << FormatIntHex(c, c < 256 ? 2 : 4);
    return s;
}
```

```
String& operator*(String& s, const String& q)
{
    for(int c : q) s * c;
    return s;
}
```

```
CONSOLE_APP_MAIN
{
    Vector<String> v;
    for(int i = 0; i < 256; i++) v.Add() * i;
    Cout() << v.ToString();
}
```

Best regards,

Oblivion

---

---

Subject: Re: operation with ascii table character (Pascal x C++/U++)

Posted by [mirek](#) on Sat, 17 Apr 2021 20:05:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Oblivion wrote on Sat, 17 April 2021 18:20Hello BetoValle,

I see. AFAIK there is no such formatter in U++ (it is too specific, IMO).

But it is easy to create one, using operator overloading, for example.

```
String& operator*(String& s, int c)
{
    if(c >= 0x20 && c <= 0x7E)
        s.Cat(c);
    else
        s << "0x" << FormatIntHex(c, c < 256 ? 2 : 4);
    return s;
}
```

```
String& operator*(String& s, const String& q)
{
    for(int c : q) s * c;
    return s;
}
```

```
CONSOLE_APP_MAIN
{
    Vector<String> v;
    for(int i = 0; i < 256; i++) v.Add() * i;
    Cout() << v.ToString();
}
```

Best regards,  
Oblivion

BTW, Format is extensible, so you can in fact add this somewhat weird formatter to Format..

---

---

Subject: Re: operation with ascii table character (Pascal x C++/U++)

Posted by [zsoft](#) on Sat, 17 Apr 2021 21:37:22 GMT

You don't need this in C++ and you can add any non printable characters to U++ strings:

```
CONSOLE_APP_MAIN
{
    String s;
    s << (char)4;
    Cout()<< "show " << s << EOL; // not show!

}
```

It will not be shown on terminal, but if you redirect the output to a file, it will be there.

./your\_program > your\_file.txt

or

your\_program.exe > your\_file.txt

---

---

Subject: Re: operation with ascii table character (Pascal x C++/U++)

Posted by [BetoValle](#) on Sat, 17 Apr 2021 22:12:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I thank you all very much. The routine was this low, of simple encryption. To decrypt the function is the same. The issue of content resulting from encryption does not need to be displayed (internally the contents are stored, for example in the database) and when decrypting the content recovery, it is correct and logically visible. Only that the function can be improved, but it is a my start.

"mStr" is content you want to encrypt

"mchave" is encryption key content

```
String Criptografia(String mStr, String mChave)
```

```
{
    String s;
    int pos,posLetra;
    mChave=mChave;
    int TamanhoString = mStr.GetLength();
    int TamanhoChave = mChave.GetLength();

    for(int i = 0; i < TamanhoString; i++){
        pos = (i % TamanhoChave);

        if( pos == 0)
            pos = TamanhoChave;
```

```
int a = mStr[i];
int b = mChave[pos];

posLetra = a ^ b;

if (posLetra == 0)
    posLetra = mStr[i];
s << char(posLetra);
}
return s;
}
```

CONSOLE\_APP\_MAIN

```
{
//first time you encrypts
String a=Criptografia("José da Silva","efg");
//second time you decrypts
String b=Criptografia(a,"efg");
    Cout()<< b << EOL;
}
```