

Lesson 49 atoi & stoi

فى هذا الدرس نريد عمل برنامج يقوم بطباعة binary الخاص بأى رقم سيكون كالتالى:

```
#include <iostream>
using namespace std;

int main() {
    int x = 5;
    for (int i = 31; i >= 0; i--)
        // قمنا بعمل for نقوم بالدوران 32
        if (x & (1 << i))
            cout << "1";
        else
            cout << "0";
    cout << endl;
}
```

output:

00000000000000000000000000000000101

لكن إذا أردنا عمل ~5

```
#include <iostream>
using namespace std;

int main() {
    int x = ~5;
```

```

for (int i = 31; i >= 0; i--)
    if (x & (1 << i))
        cout << "1";
    else
        cout << "0";
cout << endl;
}

```

output:

11111111111111111111111111111111010

وإذا أردنا طباعة الرقم السالب :

```

#include <iostream>
using namespace std;

int main() {
    int x = ~5;
    cout << x << endl;
    // طباعة الرقم السالب
    for (int i = 31; i >= 0; i--)
        if (x & (1 << i))
            cout << "1";
        else
            cout << "0";
    cout << endl;
}

```

output:

-6

11111111111111111111111111111111010

لكن ماذا إذا أردنا تحويل string إلى int :

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x = "2020";
    int y = atoi (x.c_str());
    //x.c_str() قمنا بتحويل string إلى string الموجود في لغة C
    // atoi تعمل مع لغة C
    cout << y + 1 << endl;
}
```

output:

2021

لكن إذا أردنا استخدام string الموجود في لغة ++c بدون تحويله سنستخدم stoi وسيكون كالتالى:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x = "2020";
    int y = stoi(x);
    //يمكننا استخدام stoi مع string الموجود في لغة ++c
    cout << y + 1 << endl;
}
```

output:

2021

لكن هناك مشكلة مع stoi وهو إذا كان هناك حرف في string سيحصل مشكلة في الكود:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x = "a";
    int y = stoi(x);
    cout << y + 1 << endl;
}
```

لكن مع atoi لا تحدث هذه المشكلة ويكون كالتالي:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x = "a";
    int y = atoi(x.c_str());
    cout << y + 1 << endl;
}
```

output:

1

لكن إذا أردنا حل مشكلة stoi نقوم باستخدام try و catch ووظيفتها إذا كان هناك خطأ متوقع حدوثه لكن نريد أن يستمر البرنامج فالعمل بدون أن يتوقف نضع الكود داخلهم كالتالي:

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    string x = "%$^&#";
    int y = -1;
    try {
        // نخبّر الكمبيوتر أن يحاول عمل هذا الكود
        y = stoi(x);
    }
    catch (exception e) {
```

```
cout << "Error" << endl;  
// إذا حدث خطأ أن يستمر البرنامج في العمل ولا يتوقف  
//Error لكن يقوم بطباعة  
y = -1;  
}  
if(y!=-1)  
cout << y + 1 << endl;  
}
```

output:

Error