

# COMPUTER SCIENCE : C++

## Armstrong Number in C++

Before going to write the C++ program to check whether the number is Armstrong or not, let's understand what is Armstrong number.

**Armstrong number** is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

Let's try to understand why **371** is an Armstrong number.

1.  $371 = (3*3*3)+(7*7*7)+(1*1*1)$
2. where:
3.  $(3*3*3)=27$
4.  $(7*7*7)=343$
5.  $(1*1*1)=1$
6. So:
7.  $27+343+1=371$

Let's see the C++ program to check Armstrong Number.

1. `#include <iostream.h>`
2. `using namespace std;`
3. `void main()`
4. `{`
5. `int n,r,sum=0,temp;`
6. `cout<<"Enter the Number= ";`
7. `cin>>n;`
8. `temp=n;`
9. `while(n>0)`

```
10.  {
11.   r=n%10;
12.   sum=sum+(r*r*r);
13.   n=n/10;
14.  }
15.  if(temp==sum)
16.   cout<<"Armstrong Number."<<endl;
17.  else
18.   cout<<"Not Armstrong Number."<<endl;
19.  return 0;
20.  }
```

Output:

```
Enter the Number= 371
Armstrong Number.
Enter the Number= 342
Not Armstrong Number.
```

# Sum of digits program in C++

We can write the sum of digits program in C++ language by the help of loop and mathematical operation only.

## Sum of digits algorithm

To get sum of each digit by C++ program, use the following algorithm:

- **Step 1:** Get number by user
- **Step 2:** Get the modulus/remainder of the number
- **Step 3:** sum the remainder of the number
- **Step 4:** Divide the number by 10
- **Step 5:** Repeat the step 2 while number is greater than 0.

Let's see the sum of digits program in C++.

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int n,sum=0,m;
6.     cout<<"Enter a number: ";
7.     cin>>n;
8.     while(n>0)
9.     {
10.    m=n%10;
11.    sum=sum+m;
12.    n=n/10;
13. }
14. cout<<"Sum is= "<<sum<<endl;
15. return 0;
16. }
```

Output:

```
Enter a number: 23
Sum is= 5
Enter a number: 624
Sum is= 12
```

## C++ Program to reverse number

We can reverse a number in C++ using loop and arithmetic operators. In this program, we are getting number as input from the user and reversing that number.

```
1. #include <iostream>
2. using namespace std;
3. void main()
4. {
5.     int n, reverse=0, rem;
6.     cout<<"Enter a number: ";
7.     cin>>n;
8.     while(n!=0)
9.     {
10.         rem=n%10;
11.         reverse=reverse*10+rem;
12.         n/=10;
13.     }
14.     cout<<"Reversed Number: "<<reverse<<endl;
15. }
```

Output:

```
Enter a number: 234
Reversed Number: 432
```

# C++ Program to swap two numbers without third variable

We can swap two numbers without using third variable. There are two common ways to swap two numbers without using third variable:

1. By + and -
2. By \* and /

## Using \* and /

- 1.
2. `#include <iostream.h>`
3. `using namespace std;`
4. `void main()`
5. `{`
6. `int a=5, b=10;`
7. `cout<<"Before swap a= "<<a<<" b= "<<b<<endl;`
8. `a=a*b; //a=50 (5*10)`
9. `b=a/b; //b=5 (50/10)`
10. `a=a/b; //a=10 (50/5)`
11. `cout<<"After swap a= "<<a<<" b= "<<b<<endl;`
12. `}`

## Output:

```
Before swap a= 5 b= 10
```

```
After swap a= 10 b= 5
```

## Using + and -

Let's see another example to swap two numbers using + and -.

```
1. #include <iostream.h>
2. using namespace std;
3. void main()
4. {
5.     int a=5, b=10;
6.     cout<<"Before swap a= "<<a<<" b= "<<b<<endl;
7.     a=a+b; //a=15 (5+10)
8.     b=a-b; //b=5 (15-10)
9.     a=a-b; //a=10 (15-5)
10. cout<<"After swap a= "<<a<<" b= "<<b<<endl;
11. }
```

### Output:

```
Before swap a= 5 b= 10
After swap a= 10 b= 5
```