

Day 4-31/03/2022

1. Sum of Digits Prime/Non-Prime

class Solution

{

public:

int digitPrime(int N)

{

int sum=N%9;

if(sum==2 || sum==3 || sum==5 || sum==7)

{

return 1;

}

else if(sum==1)

{

return 0;

}

else

```
{  
    return 0;  
}  
}
```

```
bool isPrime(int N)  
{  
    if(N==0 || N==1)  
    {  
        return false;  
    }  
    for(int i = 2; i <=sqrt(N); i++)  
    {  
        if(N%i==0)  
        {  
            return false;  
        }  
    }  
    return true;  
}
```

```
}
```

2.Product of Prime Factors

```
long long int primeProduct(int N)
```

```
{
```

```
    long long int product = 1;
```

```
    for(int i = 1; i <= N; i++)
```

```
    {
```

```
        if(N%i==0)
```

```
        {
```

```
            if(isPrime(i))
```

```
            {
```

```
                product=product* i;
```

```
            }
```

```
        }
```

```
    }
```

```
    return product;
```

```
}
```

3.Print the pattern

```
#include <iostream>

using namespace std;

int main()
{
    int n,count=1;
    cin>>n;
    std::cout<<"Enter the lines to be printed:\n ";
    for(int row=1;row<=n;row++)
    {

        for(int col=1;col<=row;col++)
        {
            std::cout<<count<<" ";
            count=count+2;
        }
        std::cout<<endl;
```

```
    }  
    return 0;  
}
```

4.Replace all 0's with 5

```
int convertFive(int n)  
{  
    int num=0;  
    if(n==0)  
        return 5;  
    while(n>0)  
    {  
        if(n%10==0)  
            num=num*10+5;  
        else  
            num=num*10+n%10;  
        n=n/10;  
    }  
    n=0;  
    while(num>0)
```

```
{  
    n=n*10+num%10;  
    num/=10;  
}  
return n;  
}
```

5. Pattern Jumping

```
string canJump(long long N) {  
    long long i=1;  
    while(i<N)  
    {  
        i+=i;  
    }  
    if(i==N)  
    {  
        return "True";  
    }  
    else  
    {
```

```
    return "False";
```

```
}
```

```
}
```