

Write a function that accepts a positive integer as a parameter and then returns a representation of that number in binary (base 2).

```
def base_number(number):  
    binary= 0  
    place = 1  
    for i in range(32):  
        remainder= number % 2  
        binary += remainder * place  
        place *= 10  
        number //= 2  
        if number == 0 :  
            return binary  
    return binary  
print(base_number(45))
```

101101

*# Write and test a function that takes an integer as its parameter and returns the factors of that integer.
(A factor is an integer which can be multiplied by another to yield the original)*

```
def prime_factors(number):  
    factors =[]  
    for i in range (1, number+1):  
        if number % i==0:  
            factors.append(i)  
    return factors  
print(prime_factors(10))
```

[1, 2, 5, 10]

*# Write and test a function that determines if a given integer is a prime number. A prime number is an integer greater than 1 that cannot be produced
by multiplying two other integers.*

```
def prime_factors(number):  
    factors =[]  
    for i in range(1,number+1):  
        if number % i ==0:  
            factors.append(i)  
    return factors  
print(prime_factors(20))
```

[1, 2, 4, 5, 10, 20]

Computers are commonly used in encryption. A very simple form of encryption (more accurately "obfuscation") would be to remove the

spaces from a
message and reverse the resulting string. Write, and test, a
function that takes a string containing a message and "encrypts" it in
this way .

```
def identifying_prime(number):  
    if number<=1:  
        return False  
    for number in range(2, number+1):  
        return True  
    for divisor in range(2,number):  
        if number % divisor== 0:  
            return False  
  
print(identifying_prime(20))
```

True

Another way to hide a message is to include the letters that make it
up within seemingly random text. The letters of the message might be
every
fifth character for example. Write and test a function that does such
encryption. It should randomly generate an interval (between 2 and
20), space
the message out accordingly, and should fill the gaps with random
letter

```
import random  
import string  
  
def encrypt_message(message):  
    interval = random.randint(2, 5)  
    encrypted_message = ""  
    count = 0  
  
    for char in message:  
        encrypted_message += char  
        count += 1  
        if char != " " and count % interval == 0:  
            encrypted_message += random.choice(string.ascii_lowercase)  
  
    return encrypted_message, interval  
  
message = "send cheese"  
encrypted, interval = encrypt_message(message)  
print(f"Encrypted message: {encrypted}")  
print(f"Interval used: {interval}")
```

Encrypted message: sepnds cthevesje
Interval used: 2

Write a program that decrypts messages encoded as above.

```
def encrypt_message(message):  
    encrypted = ""  
    for char in message:  
        if char != "":  
            encrypted = char + encrypted  
    return encrypted
```

```
message= "BLACK PINK"  
encrypted = encrypt_message(message)  
print(encrypted)
```

KNIP KCALB