Q1. Write a function that accepts a positive integer as a parameter and then returns a representation of that number in binary (base 2).

Ans.

Hint: This is in many ways a trick question. Think!

```
def binary(n):
   return bin(n)[2:]
print(binary(10))
```

Output:

1010

Q2. 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).

Ans.

```
def find_factors(n):
   factors = []
   for i in range(1, n + 1):
      if n % i == 0:
         factors.append(i)
   return factors
print(find_factors(12))
```

Output:

[1, 2, 3, 4, 6, 12]

Q3. 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.

Ans.

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True


print(is_prime(11))
print(is_prime(15))
print(is_prime(1))
print(is_prime(17))
```

Output:

True
False
False
True

Q4. 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.

Ans.

```
def encrypt_message(message):
    return message.replace(" ", "")[::-1]
print(encrypt_message("Hello World"))
```

Output:

dlroWolleH

Q5. 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 fi h 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 letters. The function should return the encrypted message and the interval used.

For example, if the message is "send cheese", the random interval is 2, and for clarity the random letters are not random:

send cheese

s e n d c h e e s e

sxyexynxydxy cxyhxyexyexysxye

Ans.

```
import random
import string
def encrypt_message(message):
    interval = random.randint(2, 20)
    encrypted_message = []
    for i in range(len(message)):
        if i % interval == 0:
            encrypted_message.append(message[i])
        else:
            random_letter = random.choice(string.ascii_lowercase)
            encrypted_message.append(random_letter)
    encrypted_message_str = ''.join(encrypted_message)
    return encrypted_message_str, interval
message = "send cheese"
encrypted, interval = encrypt_message(message)
print(f"Original message: {message}")
```

```
print(f"Encrypted message: {encrypted}")
print(f"Interval used: {interval}")
```

Output:

Original message: send cheese
Encrypted message: sfqzfrkxpsh
Interval used: 9

Q6. Write a program that decrypts messages encoded as above.

Ans.

```
def decrypt_message(encrypted_message, interval):
   original_message = []
   for i in range(len(encrypted_message)):
      if i % interval == 0:
         original_message.append(encrypted_message[i])
   decrypted_message = ''.join(original_message)
   return decrypted_message
encrypted_message = "sxyexynxydxycxyhxyexyexysxye"
interval = 2
decrypted = decrypt_message(encrypted_message, interval)
print(f"Encrypted message: {encrypted_message}")
print(f"Decrypted message: {decrypted}")
```

Output:

Encrypted message: sxyexynxydxycxyhxyexyexysxye
Decrypted message: syxnyxcyxeyxsy