Q1: The keyword used to create a function in Python is def. Here's a function to return a list of odd numbers in the range of 1 to 25:

```
def odd_numbers():
    return [num for num in range(1, 26) if num % 2 != 0]

# Test the function
print(odd_numbers())
```

Q2: *args and **kwargs are used in functions to accept a variable number of positional and keyword arguments, respectively.

```
# Function with *args
def sum_args(*args):
    return sum(args)

# Function with **kwargs
def print_kwargs(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")

# Test the functions
print(sum_args(1, 2, 3, 4, 5))
print_kwargs(a=1, b=2, c=3)
```

Q3: An iterator in Python is an object that represents a stream of data. The iter() method is used to initialize the iterator object, and the next() method is used for iteration.

```
# Initialize the iterator object
iter_obj = iter([2, 4, 6, 8, 10, 12, 14, 16, 18, 20])

# Iterate and print the first five elements
for _ in range(5):
    print(next(iter_obj))
```

Q4: A generator function in Python is a special type of function that generates a sequence of values using the yield keyword. The yield keyword is used to yield values one at a time, allowing the function to maintain its state between calls.

```
def generate_numbers():
    for num in range(1, 6):
        yield num

# Test the generator function
for value in generate_numbers():
    print(value)
```

Q5: Here's a generator function for prime numbers less than 1000:

```
  def generate_primes():
    for num in range(2, 1000):
        if all(num % i != 0 for i in range(2, int(num**0.5) + 1)):
            yield num

# Use the next() method to print the first 20 prime numbers
prime_gen = generate_primes()
for _ in range(20):
    print(next(prime_gen))
```

Q6:

Here's a Python program to print the first 10 Fibonacci numbers using a while loop:

```
def fibonacci():
    a, b = 0, 1
    count = 0
    while count < 10:
        print(a, end=" ")
        a, b = b, a + b
        count += 1

# Call the function to print the first 10 Fibonacci numbers
fibonacci()
```

Q7: Here's a list comprehension to iterate through the given string 'pwskills':

```
string = 'pwskills'
result = [char for char in string]
print(result)
```

Q8: Here's a Python program to check whether a given number is a Palindrome or not using a while loop:

```
def is_palindrome(num):
    original_num = num
    reverse_num = 0
    while num > 0:
        digit = num % 10
        reverse_num = reverse_num * 10 + digit
        num //= 10
    return original_num == reverse_num

# Test the function
number = int(input("Enter a number: "))
if is_palindrome(number):
    print(f"{number} is a palindrome.")
else:
    print(f"{number} is not a palindrome.")
```

Q9: Here's a code to print odd numbers from 1 to 100 using list comprehension:

```
odd_numbers = [num for num in range(1, 101) if num % 2 != 0]
print(odd_numbers)
```