```
import random
import string

def generate_password(length, uppercase=True, lowercase=True, numbers=True, symbols=True):
    characters = ''
    if uppercase:
        characters += string.ascii_uppercase
    if lowercase:
        characters += string.ascii_lowercase
    if numbers:
        characters += string.digits
    if symbols:
        characters += string.punctuation

    if not characters:
        raise ValueError("At least one character type must be enabled.")

    password = ''.join(random.choice(characters) for _ in range(length))

    # Ensure at least one character from each selected type
    if uppercase and not any(char.isupper() for char in password):
        password = password[:random.randint(0, length-1)] + random.choice(string.ascii_uppercase) + password[random.randint(0, length-1):]
    if lowercase and not any(char.islower() for char in password):
        password = password[:random.randint(0, length-1)] + random.choice(string.ascii_lowercase) + password[random.randint(0, length-1):]
    if numbers and not any(char.isdigit() for char in password):
        password = password[:random.randint(0, length-1)] + random.choice(string.digits) + password[random.randint(0, length-1):]
    if symbols and not any(char in string.punctuation for char in password):
        password = password[:random.randint(0, length-1)] + random.choice(string.punctuation) + password[random.randint(0, length-1):]

    return password

def generate_multiple_passwords(length, count, uppercase=True, lowercase=True, numbers=True, symbols=True):
    passwords = []
    for _ in range(count):
        passwords.append(generate_password(length, uppercase, lowercase, numbers, symbols))
    return passwords

# Example usage:
length = 12
count = 5
passwords = generate_multiple_passwords(length, count)
for password in passwords:
    print(password)
```