


```
import random
import string
import argparse

def generate_password(length, use_uppercase=True, use_lowercase=
True, use_digits=True, use_special_chars=True):
    characters = ''

    if use_uppercase:
        characters += string.ascii_uppercase
    if use_lowercase:
        characters += string.ascii_lowercase
    if use_digits:
        characters += string.digits
    if use_special_chars:
        characters += string.punctuation

    if not characters:
        print(
"Error: At least one character set should be selected.")
        return None
```



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

```
    return password
```

```
def generate_multiple_passwords(num_passwords, length,
use_uppercase=True, use_lowercase=True, use_digits=True,
use_special_chars=True):
```

```
    passwords = [generate_password(length, use_uppercase,
use_lowercase, use_digits, use_special_chars) for _ in range(
num_passwords)]
```

```
    return passwords
```

```
def main():
```

```
    parser = argparse.ArgumentParser(description=
'Generate strong, secure passwords.')
```

```
    parser.add_argument('-n', '--num-passwords', type=int,
default=1, help='Number of passwords to generate')
```

```
    parser.add_argument('-l', '--length', type=int, default=12,
help='Length of each password')
```

```
    parser.add_argument('--no-uppercase', dest='use_uppercase',
action='store_false', help='Exclude uppercase letters')
```

```
    parser.add_argument('--no-lowercase', dest='use_lowercase',
action='store_false', help='Exclude lowercase letters')
```

```
    parser.add_argument('--no-digits', dest='use_digits', action
='store_false', help='Exclude digits')
```

```
    parser.add_argument('--no-special-chars', dest=
'use_special_chars', action='store_false', help=
'Exclude special characters')
```

```
    args = parser.parse_args()
```

```
    num_passwords = args.num_passwords
```

```
    length = args.length
```



```
1     use_uppercase = args.use_uppercase
2     use_lowercase = args.use_lowercase
3     use_digits = args.use_digits
4     use_special_chars = args.use_special_chars
5
6     passwords = generate_multiple_passwords(num_passwords,
length, use_uppercase, use_lowercase, use_digits,
use_special_chars)
7
8     print(f'Generated {num_passwords} passwords of length {
length}:')
9     for i, password in enumerate(passwords, start=1):
10         print(f'{i}. {password}')
11
12 if __name__ == '__main__':
13     main()
```

