



```
long long factorial(int number) {  
    long long result = 1;  
    int i = 1;  
    while (i <= number) {  
        result *= i;  
        i++;  
    }  
    return result;  
}
```



```
long long factorial(int number) {  
    long long result = 1;  
    int i = 1;  
    while (i <= number) { // условие  
        result *= i;  
        i++;  
    }  
    return result;  
}
```



```
long long factorial(int number) {  
    long long result = 1;  
    int i = 1;  
    while (i <= number) { // условие  
        result *= i;  
        i++; // итерация  
    }  
    return result;  
}
```




```
long long factorial(int number) {  
    long long result = 1;  
    int i = 1; // инициализация  
    while (i <= number) { // условие  
        result *= i;  
        i++; // итерация  
    }  
    return result;  
}
```



```
int isPrime(int number) {  
    if (number <= 1) {  
        return 0;  
    }  
    int i = 2;  
    while (i * i <= number) {  
        if (number % i == 0) {  
            return 0;  
        }  
        i++;  
    }  
    return 1;  
}
```



```
int isPrime(int number) {  
    if (number <= 1) {  
        return 0;  
    }  
    int i = 2;  
    while (i * i <= number) { // условие  
        if (number % i == 0) {  
            return 0;  
        }  
        i++;  
    }  
    return 1;  
}
```



```
int isPrime(int number) {  
    if (number <= 1) {  
        return 0;  
    }  
    int i = 2;  
    while (i * i <= number) { // условие  
        if (number % i == 0) {  
            return 0;  
        }  
        i++; // итерация  
    }  
    return 1;  
}
```



```
int isPrime(int number) {  
    if (number <= 1) {  
        return 0;  
    }  
    int i = 2; // инициализация  
    while (i * i <= number) { // условие  
        if (number % i == 0) {  
            return 0;  
        }  
        i++; // итерация  
    }  
    return 1;  
}
```





```
void printBinaryRepresentation(int number) {  
    int i = 1;  
    while (i <= number) {  
        if (number & i) {  
            printf("1");  
        } else {  
            printf("0");  
        }  
        i <<= 1;  
    }  
    printf("\n");  
}
```



```
void printBinaryRepresentation(int number) {  
    int i = 1;  
    while (i <= number) { // условие  
        if (number & i) {  
            printf("1");  
        } else {  
            printf("0");  
        }  
        i <<= 1;  
    }  
    printf("\n");  
}
```



```
void printBinaryRepresentation(int number) {  
    int i = 1;  
    while (i <= number) { // условие  
        if (number & i) {  
            printf("1");  
        } else {  
            printf("0");  
        }  
        i <<= 1; // итерация  
    }  
    printf("\n");  
}
```



```
void printBinaryRepresentation(int number) {  
    int i = 1; // инициализация  
    while (i <= number) { // условие  
        if (number & i) {  
            printf("1");  
        } else {  
            printf("0");  
        }  
        i <<= 1; // итерация  
    }  
    printf("\n");  
}
```



```
for (инициализация; условие; итерация)
{
    // тело цикла
}
```



```
for (int i = 1; i <= 10; ++i)
{
    printf("%d ", i);
}
```



```
long long factorial(int number) {  
    long long result = 1;  
    int i = 1;  
    while (i <= number) {  
        result *= i;  
        i++;  
    }  
    return result;  
}
```



```
long long factorial(int number) {  
    long long result = 1;  
    for (int i = 1; i <= number; i++) {  
        result *= i;  
    }  
    return result;  
}
```