1 Moderncode

Title

```c
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
```

This title is very very very very very very very very very very long

```c
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
```

```c
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
```

This is a very long code

```c
int main(int ac, char *av[])
{
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
```

```
    printf("Hello, World");
    printf("Hello, World");
    return 0;
}
```

```
\documentclass[10pt]{article}
\usepackage{moderncode}

\begin{document}
\section{Moderncode}

\begin{moderncode}[C][adjusted title={Title}]
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
\end{moderncode}

\begin{moderncode}[C][adjusted title={This title
    ↪  is very very very very very very very
    ↪ very very long}]
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
\end{moderncode}

\begin{moderncode}[C]
int main(int ac, char *av[])
{
    printf("Hello, World");
    return 0;
}
\end{moderncode}

\begin{moderncode}[C][adjusted title={This is a
    ↪ very long code}]
int main(int ac, char *av[])
{
    printf("Hello, World");
    printf("Hello, World");
```

```
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    printf("Hello, World");
    return 0;
}
\end{moderncode}

\moderncodeinput[TeX][]{example.tex}

\subsection{Output}

\begin{moderncodeout}
Enter a positive integer: 100
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21,
   ↪ 34, 55, 89
\end{moderncodeout}

\subsection{Inline}

\subsubsection{Inline Code}

This is an inline modern code display: \
   ↪ moderncodeinline{\LaTeX}. It is also
   ↪ possible to define a language: \
   ↪ moderncodeinline[SQL]{SELECT pg_relation_
   ↪ size('title_basics');}.

\subsection{Inline Key}

It also supports a key-like-style inline element
   ↪ : \moderncodekey{Ctrl} + \moderncodekey{C}

\section{Lstlisting}
```

```
74
75 \begin{lstlisting}[caption=Example in C++,
       ↪ language=c++]
76 #include <iostream>
77 using namespace std;
78
79 int main() {
80     int n, t1 = 0, t2 = 1, nextTerm = 0;
81
82     cout << "Enter the number of terms: ";
83     cin >> n;
84
85     cout << "Fibonacci Series: ";
86
87     for (int i = 1; i <= n; ++i) {
88         // prints the first two terms.
89         if(i == 1) {
90             cout << t1 << ", ";
91             continue;
92         }
93         if(i == 2) {
94             cout << t2 << ", ";
95             continue;
96         }
97         nextTerm = t1 + t2;
98         t1 = t2;
99         t2 = nextTerm;
100
101         cout << nextTerm << ", ";
102     }
103     return 0;
104 }
105 \end{lstlisting}
106
107 \subsection{Output}
108
109 \begin{lstlisting}[style=lstoutput]
110 Enter a positive integer: 100
111 Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21,
       ↪ 34, 55, 89
112 \end{lstlisting}
113
114 \subsection{Inline}
115
```

```
\subsubsection{Inline Code}

\lstinline{\LaTeX}

\end{document}
```

1.1 Output

```
Enter a positive integer: 100
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
    ↪ 55, 89
```

1.2 Inline

1.2.1 Inline Code

This is an inline modern code display: `\LaTeX`. It is also possible to define a language: `SELECT pg_relation_size('title_basics');`.

1.3 Inline Key

It also supports a key-like-style inline element: Ctrl + C

2 Lstlisting

Listing 1: Example in C++

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, t1 = 0, t2 = 1, nextTerm = 0;

    cout << "Enter the number of terms: ";
    cin >> n;

    cout << "Fibonacci Series: ";

    for (int i = 1; i <= n; ++i) {
        // prints the first two terms.
        if(i == 1) {
            cout << t1 << ", ";
            continue;
        }
```

```
18          if(i == 2) {
19              cout << t2 << ", ";
20              continue;
21          }
22          nextTerm = t1 + t2;
23          t1 = t2;
24          t2 = nextTerm;
25
26          cout << nextTerm << ", ";
27      }
28      return 0;
29  }
```

2.1 Output

Output

```
Enter a positive integer: 100
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
    ↪ 55, 89
```

2.2 Inline

2.2.1 Inline Code

`\LaTeX`