Code :

```
package main

import “fmt”

func main() {
      cnp := make(chan func(), 10)
      for i := 0; i < 4; i++ {
            go func() {
                  for f := range cnp {
                        f()
                  }
            }()
      }
      cnp <- func() {
            fmt.Println(“HERE1”)
      }
      fmt.Println(“Hello”)
}
```

The code creates a **buffered chan with size 10**. Then it loops for four iterations each spinning a goroutine and ranging over the channel to execute functions.

More information on loop:
The loop **starts four separate goroutines**, each capable of processing functions sent to the channel. This means up to 4 functions can be executed concurrently.

Later, the **cnp channel is fed with a anonymous function** which prints a line “HERE1”
And at the end it prints the line “Hello”.

There are some issues in the code though, the function which prints the line “HERE1” will not be called as the main routine is not waiting for goroutines to finish.