Buffer Overflow- Part B

In this lab, you need to exploit the following program by using a buffer overflow attack to run some code of choice.

```
#include <stdio.h>
int main(int argc, char * argv[]) {
    char buffer[x];
    if (argc < 2) {
        printf("usage: %s buffer
", argv[0]);
        exit(0);
    }
    strcpy(buffer, argv[1]);
    printf("ur buffer: %s", buffer);
}
```

In order to execute raw exploit codes directly in the stack or other parts of the memory, which deal with binary, we need assembly codes that represent a raw set of machine instructions of the target machines. These codes (shellcode) have to be loaded into the memory so that we can force the vulnerable program to jump to it.

Your task is to use the following shell code for the exploit. Identify the least value of the buffer length x that would make it possible to fit the shellcode in the stack? If the buffer length x=10, what strategy could be used to enable the exploit using this shellcode? What does this indicate about the desirable properties for shellcodes?

```
char shellcode[] =
```
 Also, describe what does shell code does.

**Submission instructions (For each part)**

- Detailed Log of your attack describing all the commands used as well as any intermediate values found and finally a clarification of what the given shellcode does.

- Screenshots that show you have succeeded.

- Provide brief concise answers to all the questions related to the buffer length and the purpose of the shellcode.

- Use 'script' to record your session.