What is the output of the following codes?

```c
#include <stdio.h>
#include <unistd.h>
int main()
{
    if (fork() || fork())
        fork();
    printf("1 ");
    return 0;
}
```
//

```c
#include <stdio.h>
#include <unistd.h>

int main()
{
    if (fork() && (!fork())) {
        if (fork() || fork()) {
            fork();
        }
    }
    printf("2 ");
    return 0;
}
```
//

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{

    // make two process which run same
    // program after this instruction
    fork();

    printf("Hello world!\n");
    return 0;
}
```
//

```c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
    fork();
    fork();
    printf("hello\n");
    return 0;
}
```
//

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
void forkexample()
{
    // child process because return value zero
    if (fork() == 0)
         printf("Hello from Child!\n");

    // parent process because return value non-zero.
    else
         printf("Hello from Parent!\n");
}
int main()
{
    forkexample();
    return 0;
}
```
//
```c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
printf("Ali!\n");
    fork();
printf("Ahmed!\n");
    fork();
    printf("hello\n");
    return 0;
}
```
//
```c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
fork();
printf("Ali!\n");
    fork();
fork();
printf("Ahmed!\n");
    fork();
    printf("hello\n");
    return 0;
}
```
//
```c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
    fork();
```

```c
    fork();
    fork();
    printf("hello\n");
    return 0;
}
```
//

```c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
    fork();
    fork();
    printf("hello\n");
    fork();
    fork();
    printf("Pakistan")
    return 0;
}
```