

۱. خروجی برنامه‌ی زیر را مشخص کنید.

```
2. int main() {
3.     printf("L0\n");
4.     p=fork();
5.     printf("L1\n");
6.     p=fork();
7.     printf("Bye\n");
8. }
```

۲. خروجی دستور زیر پس از اجرای برنامه C نشان داده شده چه تغییری خواهد کرد؟

\$ ps -e -o pid,ppid,stat,cmd

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    pid_t child_pid;
    child_pid=fork();
    if(child_pid>0){
        sleep(60);
    }

    else{
        exit(0);
    }

    return 0;
}
```

۳. عملکرد برنامه‌ی زیر را توضیح دهید.

```
4. int spawn(char* program,char** arg_list)
5. {
6.     pid_t child_pid;
7.     child_pid=fork();
8.     if(child_pid!=0)
9.         return child_pid;
10.    else{
11.        execvp(program,arg_list);
12.        fprintf(stderr,"an error occurred in execvp\n");
13.        abort();
14.    }
```

```

14.         }
15.     }
16.
17.     int main()
18.     {
19.         char* arg_list[]={
20.             "ls"
21.             "-l",
22.             "",
23.             NULL
24.         };
25.
26.         spawn("ls",arg_list);
27.         printf("done with main program\n");
28.         return 0;
29. }

```

۴. برنامه‌ی زیر را به‌گونه‌ای تغییر دهید که پدر یک فرزند را ایجاد و از طریق pipe با یکدیگر ارتباط برقرار کنند.

```

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

void writer(const char* message,int count,FILE* stream)
{
    for(;count>0;--count){
        fprintf(stream,"%s\n",message);
        fflush(stream);
        sleep(1);
    }
}

void reader(FILE* stream)
{
    char buffer[1024];
    while(!feof(stream) && !ferror(stream) &&
fgets(buffer,sizeof(buffer),stream)!=NULL)
        fputs(buffer,stdout);
}

int main()
{
    int fds[2];
    pid_t pid;

    return 0;
}

```