

```

// P1.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;
}

// P2.c
#include <stdio.h>
#include <sys/types.h>
int main()
{
    fork();
    fork();
    fork();
    printf("hello\n");
    return 0;
}

// P3.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    int x = 1;
    if (fork() == 0)
        printf("Child has x = %d\n", ++x);
    else
        printf("Parent has x = %d\n", --x);
    return 0;
}

```

```

// p4.c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("I am: %d\n", (int) getpid());

    pid_t pid = fork();
    printf("fork returned: %d\n", (int) pid);

    if (pid < 0) { /* error occurred */
        perror("Fork failed");
    }
    if (pid == 0) { /* child process */
        printf("I am the child with pid %d\n", (int)
getpid());
        printf("Child process is exiting\n");
        exit(0);
    }
    /* parent process */
    printf("I am the parent waiting for the child
process to end\n");
    wait(NULL);
    printf("parent process is exiting\n");
    return(0);
}

```