**Q1)**#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

int main() {

int pipe1[2];

int pipe2[2];

pid\_t pid;

if (pipe(pipe1) == -1 || pipe(pipe2) == -1) {

perror("pipe");

exit(1);

}

pid = fork();

if (pid == -1) {

perror("fork");

exit(1);

}

if (pid == 0) {

close(pipe1[1]);

close(pipe2[0]);

char receivedStr[100];

read(pipe1[0], receivedStr, sizeof(receivedStr));

close(pipe1[0]);

char additionalStr[] = " World!";

int i = 0, j = 0;

while (receivedStr[i] != '\0') {

i++;

}

while (additionalStr[j] != '\0') {

receivedStr[i++] = additionalStr[j++];

}

receivedStr[i] = '\0';

write(pipe2[1], receivedStr, sizeof(receivedStr));

close(pipe2[1]);

exit(0);

} else {

close(pipe1[0]);

close(pipe2[1]);

char inputStr[100];

printf("Enter a string: ");

fgets(inputStr, sizeof(inputStr), stdin);

inputStr[strcspn(inputStr, "\n")] = '\0';

write(pipe1[1], inputStr, sizeof(inputStr));

close(pipe1[1]);

char finalStr[100];

read(pipe2[0], finalStr, sizeof(finalStr));

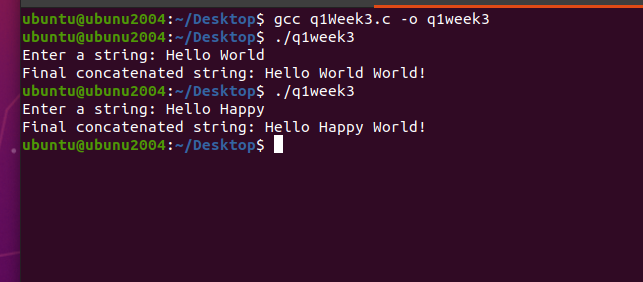
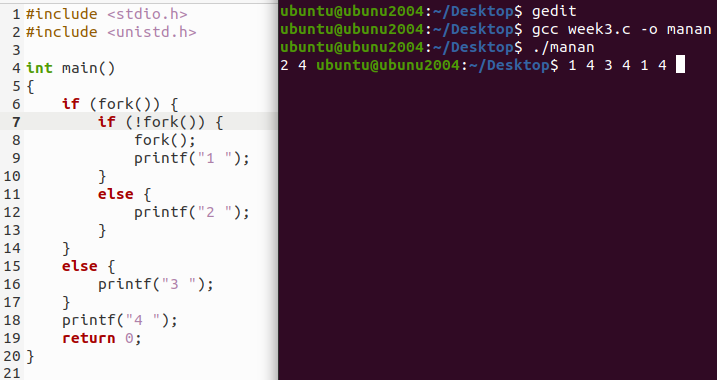
close(pipe2[0]);

printf("Final concatenated string: %s\n", finalStr);

}

return 0;

}

  
  
  
 **Q2)** ‘

**Q3)**

#include <stdio.h>

#include <unistd.h>

int main()

{

float n;

printf("Enter seconds to sleep for: ");

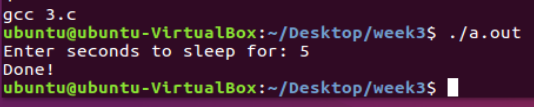
scanf("%f", &n);

sleep(n);

printf("Done!\n");

return 0;

}



**Q4)**  
)#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/wait.h>

#define NUM\_CHILDREN 4

int main() {

pid\_t pids[NUM\_CHILDREN];

int i;

for (i = 0; i < NUM\_CHILDREN; i++) {

pids[i] = fork();

if (pids[i] < 0) {

perror("fork");

exit(1);

} else if (pids[i] == 0) { // Child process

FILE \*file = fopen("my file.txt", "r");

if (file == NULL) {

perror("fopen");

exit(1);

}

char line[100];

for (int j = 0; j <= i; j++) {

if (fgets(line, sizeof(line), file) == NULL) {

perror("fgets");

fclose(file);

exit(1);

}

}

printf("Child %d reads: %s", i + 1, line);

fclose(file);

exit(0);

}

}

for (i = 0; i < NUM\_CHILDREN; i++) {

wait(NULL);

}

return 0;

}

