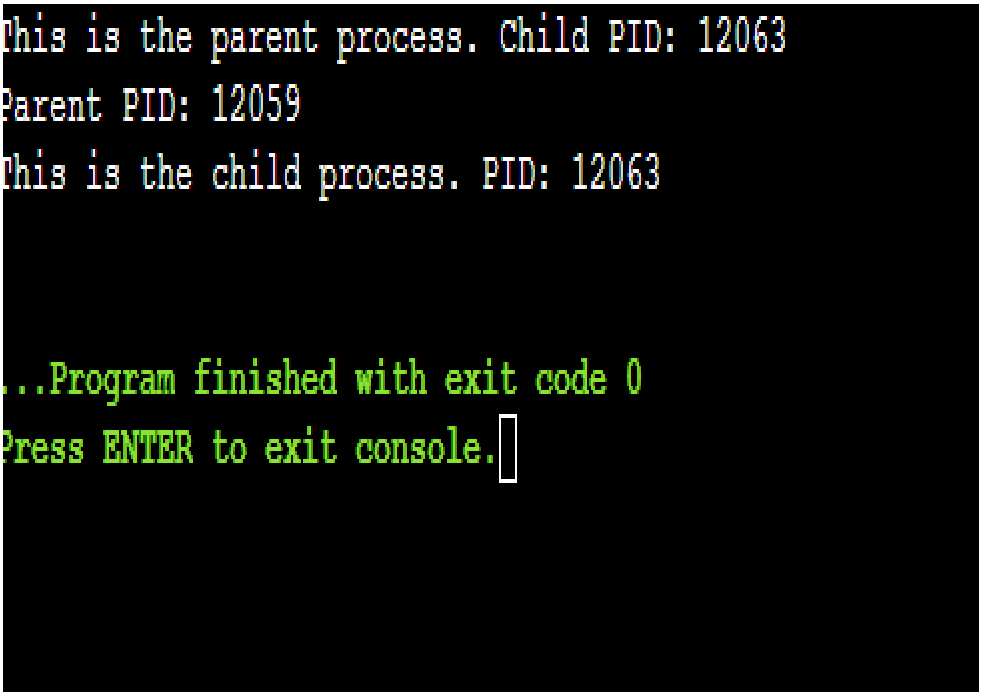
Week:1

Name: Abhinav Saini
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

1.Write a program to create a child process using system call fork().

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

int main() {
   pid_t pid = fork();

   if (pid < 0) {
      fprintf(stderr, "Fork Failed\n");
      return 1;
   }
   else if (pid == 0) {
      printf("This is the child process. PID: %d\n", getpid());
   }
   else {
      printf("This is the parent process. Child PID: %d\n", pid);
      printf("Parent PID: %d\n", getpid());
   }

   return 0;
}
```

OUTPUT:

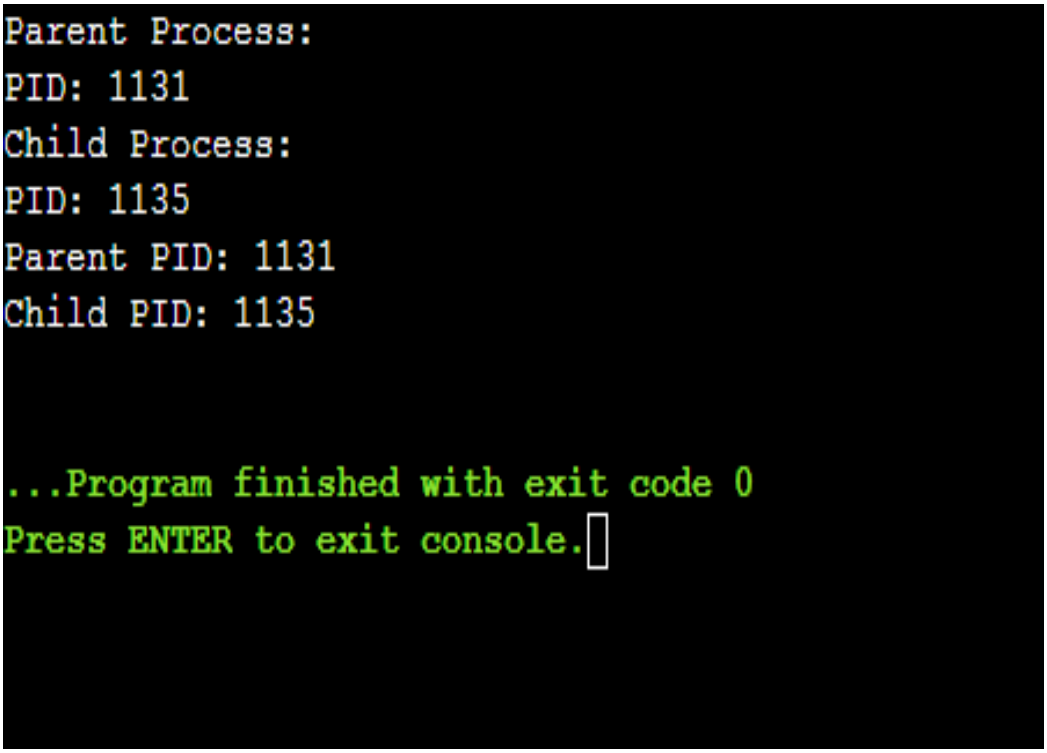

This is the parent process. Child PID: 12063
Parent PID: 12059
This is the child process. PID: 12063
...Program finished with exit code 0
Press ENTER to exit console.

Name: Mohit Bisht
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

2.Write a program to print process Id's of parent and child process i.e. parent should print its own and its child process id while child process should print its own and its parent process id. (use getpid(), getppid())

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

int main() {
   pid_t pid = fork();

   if (pid < 0) {
      fprintf(stderr, "Fork Failed\n");
      return 1;
   }
   else if (pid == 0) {

      printf("Child Process:\n");
      printf("PID: %d\n", getpid());
      printf("Parent PID: %d\n", getppid());
   }
   else {

      printf("Parent Process:\n");
      printf("PID: %d\n", getpid());
      printf("Child PID: %d\n", pid);
   }

   return 0;
}
```

OUTPUT:


Parent Process:
PID: 1131
Child Process:
PID: 1135
Parent PID: 1131
Child PID: 1135
...Program finished with exit code 0
Press ENTER to exit console.

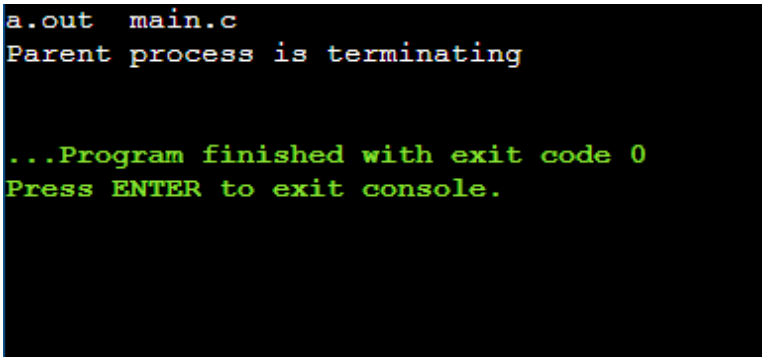
Name: Mohit Bisht
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

3. Write a program to create child process which will list all the files present in your system. Make sure that parent process waits until child has not completed its execution. (use wait(), exit()) What will happen if parent process dies before child process? Illustrate it by creating one more child of parent process.

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

int main() {
   pid_t pid1, pid2;

   pid1 = fork();

   if (pid1 < 0) {
      fprintf(stderr, "Fork Failed\n");
      return 1;
   } else if (pid1 == 0) {
      execlp("/bin/ls", "ls",
      NULL); exit(0);
   } else {
      wait(NULL);

      pid2 = fork();

      if (pid2 < 0) {
         fprintf(stderr, "Fork Failed\n");
         return 1;
      } else if (pid2 == 0) {
         sleep(5);
```

```
            printf("Second child, PID: %d, Parent PID: %d\n", getpid(),
            getppid()); exit(0);
        } else {
            printf("Parent process is terminating\n");
            exit(0);
        }
    }

    return 0;
}
```

OUTPUT:


a.out main.c
Parent process is terminating
...Program finished with exit code 0
Press ENTER to exit console.

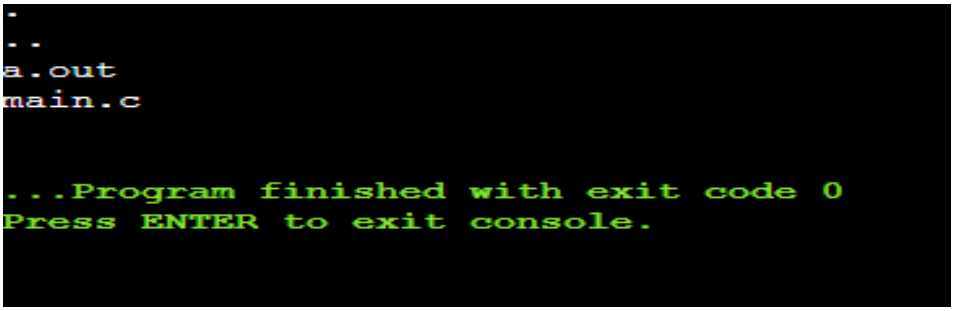
Week:2

Name: Mohit Bisht
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

4.Write a program to open a directory and list its contents. (use opendir(), readdir(), closedir())

```
#include <stdio.h>
#include <dirent.h>
int main() {
   DIR *dir;
   struct dirent *entry;

   dir = opendir(".");

   if (dir == NULL) {
      perror("Unable to open directory");
      return 1;
   }

   while ((entry = readdir(dir)) != NULL)
      { printf("%s\n", entry->d_name);
   }

   closedir(dir);

   return 0;
}
```

OUTPUT:

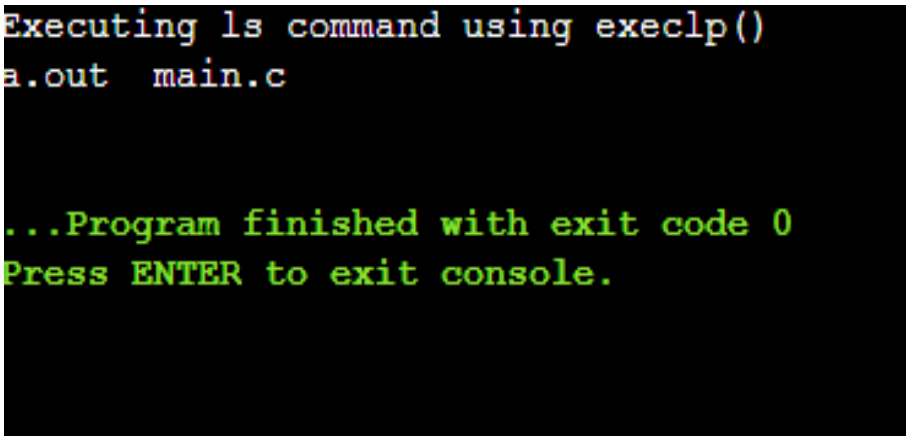

.
..
a.out
main.c
...Program finished with exit code 0
Press ENTER to exit console.

Name: Mohit Bisht
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

5.Write a program to show working of execlp() system call by executing ls command

```
#include <stdio.h>
#include <unistd.h>
int main() {
   printf("Executing ls command using execlp()\n");

   execlp("ls", "ls", NULL);

   printf("This line will not be executed if execlp is successful\n");

   return 0;
}
```

OUTPUT:


Executing ls command using execlp()
a.out main.c
...Program finished with exit code 0
Press ENTER to exit console.

Name: Mohit Bisht
Section: J1
Roll No: 37
Course: B.tech 5th Sem
Branch: CSE

6.Write a program to read a file and store your details in that file. Your program should also create one more file and store your friends details in that file. Once both files are created, print lines which are matching in both files

```
#include <stdio.h>
#include
<string.h>

#define MAX_LINE_LENGTH 256

void write_to_file(const char *filename, const char *content) {
   FILE *file = fopen(filename, "w");
   if (file == NULL) {
      perror("Unable to open file");
      return;
   }
   fprintf(file, "%s", content);
   fclose(file);
}

void find_matching_lines(const char *file1, const char *file2)
   { char line1[MAX_LINE_LENGTH],
   line2[MAX_LINE_LENGTH]; FILE *fp1 = fopen(file1,
   "r");
   FILE *fp2 = fopen(file2, "r");

   if (fp1 == NULL || fp2 == NULL) {
      perror("Error opening files");
      return;
   }

   while (fgets(line1, MAX_LINE_LENGTH, fp1) !=
      NULL) { rewind(fp2);
      while (fgets(line2, MAX_LINE_LENGTH, fp2) != NULL) {
```

```
            if (strcmp(line1, line2) == 0) {
                printf("Matching line: %s", line1);
            }
        }
    }

    fclose(fp1);
    fclose(fp2);
}

int main() {
    const char *my_details = "Name: John\nAge: 25\nCity: New York\n";
    const char *friend_details = "Name: Jane\nAge: 25\nCity: New York\n";

    write_to_file("my_details.txt", my_details);
    write_to_file("friend_details.txt", friend_details);

    printf("Matching lines in both files:\n");
    find_matching_lines("my_details.txt", "friend_details.txt");

    return 0;
}
```

OUTPUT:

main.c | friend_details.txt | my_details.txt

```
1  Name: Jane
2  Age: 25
3  City: New York
4
```

main.c | friend_details.txt | my_details.txt

```
1  Name: John
2  Age: 25
3  City: New York
4
```

```
Matching lines in both files:
Matching line: Age: 25
Matching line: City: New York


...Program finished with exit code 0
Press ENTER to exit console.
```