Write a program for following 1) zombie process 2),orphan processes 3)sum of even numbers of an array in parent and odd numbers of an array in child process

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

void create\_zombie\_process() {

pid\_t pid = fork();

if (pid > 0) {

printf("Parent process (PID: %d) created a zombie process (PID: %d)\n", getpid(), pid);

sleep(5); // Wait to keep the child as a zombie for a while

} else if (pid == 0) {

printf("Child process (PID: %d) exiting to become a zombie\n", getpid());

exit(0); // Child exits immediately, making it a zombie

} else {

perror("Fork failed for zombie process");

}

}

void create\_orphan\_process() {

pid\_t pid = fork();

if (pid > 0) {

printf("Parent process (PID: %d) exiting, creating an orphan process\n", getpid());

exit(0); // Parent exits, making the child an orphan

} else if (pid == 0) {

sleep(5); // Keep the child alive to observe orphan status

printf("Orphan child process (PID: %d) now adopted by init process (PPID: %d)\n", getpid(), getppid());

} else {

perror("Fork failed for orphan process");

}

}

void calculate\_even\_odd\_sum(int arr[], int n) {

pid\_t pid = fork();

if (pid > 0) { // Parent process calculates the sum of even numbers

int even\_sum = 0;

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

if (arr[i] % 2 == 0) {

even\_sum += arr[i];

}

}

wait(NULL); // Wait for child to finish

printf("Parent process (PID: %d) - Sum of even numbers: %d\n", getpid(), even\_sum);

} else if (pid == 0) { // Child process calculates the sum of odd numbers

int odd\_sum = 0;

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

if (arr[i] % 2 != 0) {

odd\_sum += arr[i];

}

}

printf("Child process (PID: %d) - Sum of odd numbers: %d\n", getpid(), odd\_sum);

exit(0); // Exit child process

} else {

perror("Fork failed for even-odd sum calculation");

}

}

int main() {

int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int n = sizeof(arr) / sizeof(arr[0]);

// Task 1: Create a zombie process

create\_zombie\_process();

sleep(6); // Wait for zombie to be cleared

// Task 2: Create an orphan process

create\_orphan\_process();

sleep(6); // Wait to observe orphan process

// Task 3: Calculate sum of even and odd numbers using parent and child process

calculate\_even\_odd\_sum(arr, n);

return 0;

}