

### Zombie.c

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

void bubblesort(int arr[], int n) {
    int i, j, temp;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void insertionsort(int arr[], int n) {
    int i, j, temp;
    for (i = 1; i < n; i++) {
        temp = arr[i];
        j = i - 1;
        while (j >= 0 && temp < arr[j]) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = temp;
    }
}

int main() {
    int arr[25], n, i, status;

    printf("Enter the number of values in array: ");
    scanf("%d", &n);
    printf("Enter the array elements: ");
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("\n--- Starting Fork ---\n");
    pid_t pid = fork();

    if (pid < 0) {
        perror("fork failed");
        exit(1);
    }
```

```

else if (pid == 0) {
    printf("\nCHILD: My process ID is %d\n", getpid());
    printf("CHILD: My parent's ID is %d\n", getppid());

    insertionsort(arr, n);
    printf("CHILD: Sorted using insertion sort: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\nCHILD: Exiting.\n");
    exit(0); // explicitly exit child here
}

else {
    printf("\nPARENT: My process ID is %d\n", getpid());
    printf("PARENT: My child's ID is %d\n", pid);

    printf("PARENT: Sleeping for 10 seconds. Check 'ps -l' to see
zombie.\n");
    sleep(10);

    wait(&status);

    printf("PARENT: Child has been reaped. Parent is now exiting.\n");
}
}

}

```

#### OUTPUT:-

```

ict@mplab-12:~/Desktop/33164$ gcc zombie.c -o zombie
ict@mplab-12:~/Desktop/33164$ ./zombie
Enter the number of values in array: 5
Enter the array elements: 6
7
4
8
1

--- Starting Fork ---

PARENT: My process ID is 8641
PARENT: My child's ID is 8687
PARENT: Sleeping for 10 seconds. Check 'ps -l' to see zombie.

```

```
CHILD: My process ID is 8687
CHILD: My parent's ID is 8641
CHILD: Sorted using insertion sort: 1 4 6 7 8
CHILD: Exiting.
PARENT: Child has been reaped. Parent is now exiting.
```