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

void mergeSort(int arr[], int l, int r);
void merge(int arr[], int l, int m, int r);

void merge(int arr[], int l, int m, int r) {

  int i, j, k;
  int n1 = m - l + 1;
  int n2 = r - m;

  int L[n1], R[n2];

  for (i = 0; i < n1; i++) {
    L[i] = arr[l + i];
  }
  for (j = 0; j < n2; j++) {
    R[j] = arr[m + 1 + j];
  }

  i = 0;
  j = 0;
  k = l;

  while (i < n1 && j < n2) {
    if (L[i] <= R[j]) {
      arr[k] = L[i];
      i++;
    }
    else {
      arr[k] = R[j];
      j++;
    }
    k++;
  }

  while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
  }

  while (j < n2) {
    arr[k] = R[j];
    j++;
```

```
    k++;
  }
}

void mergeSort(int arr[], int l, int r) {

  if (l < r) {

    int m = l + (r - l) / 2;

    mergeSort(arr, l, m);
    mergeSort(arr, m + 1, r);
    merge(arr, l, m, r);
  }
}

int main() {

  pid_t pid;

  int *arr,size,ch;

  printf("\nEnter size: ");
  scanf("%d",&size);
  arr=(int*)malloc(sizeof(int)*size);

  printf("\nEnter the numbers: ");
  for(int i=0;i<size;i++) {
    scanf("%d",&arr[i]);
  }

  printf("\n=====Select Option=====\n");
  printf("\n0. Synchronize Parent and Child\n1. Demonstrate Orphan\n2.
Demonstrate Zombie\n\nYour Choice: ");
  scanf("%d",&ch);

  pid = fork();

  if(pid==0) {

    printf("\n======In Child======\n\n");
    printf("I am child %d, my parent is %d\n\n",getpid(),getppid());

    printf("Array: ");
    for(int i=0;i<size;i++) {
      printf("%d ",arr[i]);
    }
    printf("\n\n");
```

```
    mergeSort(arr,0,size-1);

    printf("Merge Sorted: ");
    for(int i=0;i<size;i++) {
      printf("%d ",arr[i]);
    }
    printf("\n\n");

    if(ch==0) {
      exit(0);
    }

    else if(ch==1) {

      sleep(10);
      printf("\n\nChild %d Becomes Orphan, Adopted by Parent
%d\n\n",getpid(),getppid());
      system("ps -al");
      printf("\n");
    }

    else if(ch==2) {
      printf("Child %d Becomes Zombie\n\n",getpid());
    }
  }

  else if(pid>=1) {

    printf("\n======In Parent======\n\n");

    printf("ID: %d\n\n",getpid());

    printf("Array: ");
    for(int i=0;i<size;i++) {
      printf("%d ",arr[i]);
    }
    printf("\n\n");

    mergeSort(arr,0,size-1);

    printf("Merge Sorted: ");
    for(int i=0;i<size;i++) {
      printf("%d ",arr[i]);
    }
    printf("\n\n");

    if(ch==0) {
```

```
      int stat_val;
      pid_t child_pid;

      child_pid = wait(&stat_val);

      printf("Child has finished: PID = %d\n\n", child_pid);
      printf("======Back in Parent======\n\n");
      printf("ID: %d\n\n",getpid());
    }

    if(ch==2) {

      sleep(10);
      system("ps -al");
      printf("\n");
    }
  }
  exit(0);
}
```

```
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ gcc Assignment2a.c -o Assignment2a
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ ./Assignment2a

Enter size: 5

Enter the numbers: 2 4 1 3 5

=====Select Option=====

0. Synchronize Parent and Child
1. Demonstrate Orphan
2. Demonstrate Zombie

Your Choice: 0

======In Parent======

ID: 5026

Array: 2 4 1 3 5

Merge Sorted: 1 2 3 4 5


======In Child======

I am child 5028, my parent is 5026

Array: 2 4 1 3 5

Merge Sorted: 1 2 3 4 5

Child has finished: PID = 5028

======Back in Parent======

ID: 5026

Someshwars-MacBook-Pro:Assignments someshwargaikwad$
```

```
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ gcc Assignment2a.c -o Assignment2a
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ ./Assignment2a

Enter size: 5

Enter the numbers: 3 2 1 5 4

=====Select Option=====

0. Synchronize Parent and Child
1. Demonstrate Orphan
2. Demonstrate Zombie

Your Choice: 1

======In Parent======

ID: 5252

Array: 3 2 1 5 4

Merge Sorted: 1 2 3 4 5


======In Child======

I am child 5257, my parent is 1

Array: 3 2 1 5 4

Merge Sorted: 1 2 3 4 5

Someshwars-MacBook-Pro:Assignments someshwargaikwad$

Child 5257 Becomes Orphan, Adopted by Parent 1

  UID   PID  PPID        F CPU PRI NI       SZ    RSS WCHAN     S             ADDR TTY           TIME CMD
    0  3207  3205     4106   0  31  0  4340772   4880 -       Ss                 0 ttys000    0:00.06 login -pf someshwargaikwad
  501  3208  3207     4006   0  31  0  4329676   3224 -       S+                 0 ttys000    0:00.22 -bash
  501  5257     1        6   0  31  0  4277400    400 -       S                  0 ttys000    0:00.00 ./Assignment2a
    0  5261  5257     4106   0  31  0  4278060   1024 -       R                  0 ttys000    0:00.01 ps -al
```

```
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ gcc Assignment2a.c -o Assignment2a
Someshwars-MacBook-Pro:Assignments someshwargaikwad$ ./Assignment2a

Enter size: 5

Enter the numbers: 3 2 1 5 4

=====Select Option=====

0. Synchronize Parent and Child
1. Demonstrate Orphan
2. Demonstrate Zombie

Your Choice: 2

======In Parent======

ID: 5267

Array: 3 2 1 5 4

Merge Sorted: 1 2 3 4 5


======In Child======

I am child 5270, my parent is 5267

Array: 3 2 1 5 4

Merge Sorted: 1 2 3 4 5

Child 5270 Becomes Zombie

  UID   PID  PPID        F CPU PRI NI       SZ    RSS WCHAN     S             ADDR TTY           TIME CMD
    0  3207  3205     4106   0  31  0  4340772   4880 -       Ss                 0 ttys000    0:00.06 login -pf someshwargaikwad
  501  3208  3207     4006   0  31  0  4329676   3224 -       S                  0 ttys000    0:00.23 -bash
  501  5267  3208     4006   0  31  0  4270232    752 -       S+                 0 ttys000    0:00.00 ./Assignment2a
  501  5270  5267     2006   0   0  0        0      0 -       Z+                 0 ttys000    0:00.00 (Assignment2a)
    0  5274  5267     4106   0  31  0  4268844   1008 -       R+                 0 ttys000    0:00.00 ps -al

Someshwars-MacBook-Pro:Assignments someshwargaikwad$
```