**CODE :**

#include<stdio.h>

#include<stdlib.h>

#include<sys/types.h>

#include<unistd.h>

#include<sys/wait.h>

void arrayStore(char\* c\_arr[],int\* i\_arr, int n)

{

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

{

i\_arr[i] = atoi(c\_arr[i+1]);

}

}

void arraySort(int\* arr, int n, int order)

{

for(int i=0;i<n-1;i++)

{

for(int j=0;j<n-i-1;j++)

{

if(order == -1)

{

if(arr[j]<arr[j+1])

{

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

else if(order == 1)

{

if(arr[j]>arr[j+1])

{

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

}

void arrayDisplay(int\* arr, int n)

{

printf("\nDisplaying Array.... \n");

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

{

printf("%d\t",arr[i]);

}

printf("\n\n");

}

void main(int argc, char\* argv[])

{

int order;

int arr[argc-1];

int len = argc - 1;

arrayStore(argv,arr,len);

pid\_t pid, tpid, status;

pid = fork();

if(pid == 0)

{

//sleep(5); //Orphan State

printf("\nChild Process with ID %d\n",getpid());

printf("Child's Parent ID %d\n",getppid());

order = -1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

else

{

//sleep(5); //Zombie State

system("ps");

tpid = wait(&status);

printf("\nParent Process with ID %d \n",getpid());

order = 1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

}

**Output :**

┌──(pranav㉿Pranav)-[~]

└─$ cd OS

┌──(pranav㉿Pranav)-[~/OS]

└─$ gcc a3.c

┌──(pranav㉿Pranav)-[~/OS]

└─$ ./a.out 5 43 -34 0 -32 2

Child Process with ID 3423

Child's Parent ID 3422

Displaying Array....

43 5 2 0 -32 -34

PID TTY TIME CMD

3322 pts/0 00:00:00 zsh

3422 pts/0 00:00:00 a.out

3423 pts/0 00:00:00 a.out

3424 pts/0 00:00:00 sh

3425 pts/0 00:00:00 ps

Parent Process with ID 3422

Displaying Array....

-34 -32 0 2 5 43

**Orphan State :**

**Code:**

void main(int argc, char\* argv[])

{

int order;

int arr[argc-1];

int len = argc - 1;

arrayStore(argv,arr,len);

pid\_t pid, tpid, status;

pid = fork();

if(pid == 0)

{

sleep(5); //Orphan State

printf("\nChild Process with ID %d\n",getpid());

printf("Child's Parent ID %d\n",getppid());

order = -1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

else

{

//sleep(5); //Zombie State

system("ps");

tpid = wait(&status);

printf("\nParent Process with ID %d \n",getpid());

order = 1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

}

PID TTY TIME CMD

3894 pts/1 00:00:01 zsh

4391 pts/1 00:00:00 a.out

4352 pts/1 00:00:00 a.out

4353 pts/1 00:00:00 sh

4354 pts/1 00:00:00 ps

PID TTY TIME CMD

3894 pts/1 00:00:01 zsh

4351 pts/1 00:00:00 a.out

4352 pts/1 00:00:00 a.out

4355 pts/1 00:00:00 sh

4356 pts/1 00:00:00 ps

Child Process with ID 4352

Child's Parent ID 4391

Displaying Array....

53 34 5 0 -2 -54

Parent Process with ID 4351

Displaying Array....

-54 -2 0 5 34 53

**Zombie State :**

**Output :**

void main(int argc, char\* argv[])

{

int order;

int arr[argc-1];

int len = argc - 1;

arrayStore(argv,arr,len);

pid\_t pid, tpid, status;

pid = fork();

if(pid == 0)

{

//sleep(5); //Orphan State

printf("\nChild Process with ID %d\n",getpid());

printf("Child's Parent ID %d\n",getppid());

order = -1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

else

{

sleep(5); //Zombie State

system("ps");

tpid = wait(&status);

printf("\nParent Process with ID %d \n",getpid());

order = 1;

arraySort(arr,len,order);

arrayDisplay(arr,len);

}

}

┌──(pranav㉿Pranav)-[~/OS]

└─$ ./a.out 5 53 -54 34 -2 0

PID TTY TIME CMD

3894 pts/1 00:00:01 zsh

4298 pts/1 00:00:00 a.out

4299 pts/1 00:00:00 a.out

4300 pts/1 00:00:00 sh

4301 pts/1 00:00:00 ps

Child Process with ID 4299

Child's Parent ID 4298

Displaying Array....

53 34 5 0 -2 -54

PID TTY TIME CMD

3894 pts/1 00:00:01 zsh

4298 pts/1 00:00:00 a.out

4299 pts/1 00:00:00 a.out <defunct>

4302 pts/1 00:00:00 sh

4303 pts/1 00:00:00 ps

Parent Process with ID 4298

Displaying Array....

-54 -2 0 5 34 53