**Experiment No:2**

**zombie.c**

#include<sys/types.h>

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

int compare(const void\* num1,const void\* num2)

{

int a = \*(int\*)num1;

int b = \*(int\*)num2;

if(a > b)

{

return 1;

}else if(a < b){

return -1;

}else {

return 0;

}

}

int main()

{

int a[8],i;

printf("Enter integers :\n");

for(i=0; i<8;i++){

printf("Enter integer:%d\n",i+1);

scanf("%d",&a[i]);

}

qsort(a,8,sizeof(int),compare);

char str[128];

int j=0;

int index=0;

for(j=0;j<8;j++){

index += sprintf(&str[index],"%d\n",a[j]);

}

char \*cmd = "./bina";

char \*argu[3] = {"./bina",str,NULL};

pid\_t pid;

pid=fork();

if(pid<0){

fprintf(stderr,"Failed Fork");

}else if(pid == 0){

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

sleep(5);

}else if(pid > 0){

printf("Parent Process:%d\n",getppid());

execvp(cmd,argu);

sleep(5);

}

return 0;

}

**bina.c**

#include<stdio.h>

int binarySearch(int arr[],int l,int r,int x)

{

while(l <= r){

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

if(arr[m] == x)

return m;

if(arr[m] < x)

l = m+1;

else

r = m-1;

}

return -1;

}

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

{

const char \*str = argv[1];

printf("%s",str);

printf("\n");

int arr[8];

int numread = sscanf(str,"%d %d %d %d %d %d %d %d", &arr[0], &arr[1], &arr[2], &arr[3], &arr[4], &arr[5], &arr[6], &arr[7]);

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

int x;

printf("Enter element to be search");

scanf("%d",&x);

int result = binarySearch(arr,0,n-1,x);

(result == -1)?printf("Element is not present in array"):printf("Element is present at index %d",result);

return 0;

}

**Output:**

mml@mml-Vostro-3470:~/DSS71$ gcc bina.c -o bina

mml@mml-Vostro-3470:~/DSS71$ gcc zombie.c -o zom

mml@mml-Vostro-3470:~/DSS71$ ./zom

Enter integers :

Enter integer:1

23

Enter integer:2

32

Enter integer:3

45

Enter integer:4

12

Enter integer:5

10

Enter integer:6

13

Enter integer:7

55

Enter integer:8

9

Parent Process:4295

Child Process:5344

9

10

12

13

23

32

45

55

Enter element to be search23

Element is present at index 4