TE IT

Name : Omkar Gurav

Roll no : 8048

Implement the C program in which main program accepts an integer array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an integer array using Bubble Sort and passes the sorted array to child process through the command line argument of EXECVE system call. The child process uses EXECVE system call to load new program that uses this sorted array for performing the binary search to search the particular item in the array.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

void swap(int *x,int *y)
{
 *x=*x+*y;
 *y=*x-*y;
 *x=*x-*y;
}

void bubble(int arr[],int n)
{
 for(int i=0;i<n-1;i++)
 for(int j=0;j<n-i-1;j++)
 if(arr[j] > arr[j+1])
 swap(&arr[j], &arr[j+1]);
```

```
}

void display(int arr[],int n)
{
 for(int i=0;i<n;i++)
 printf("%d\t",arr[i]);

 printf("\n");
}

int main(int argc,char* argv[])
{
 pid_t p_id;
 int num[20],count,key,i;
 char* argp[10];

 printf("\nEnter no. of integers to be sorted: ");
 scanf("%d",&count);

 printf("\nEnter integers\n");

 for(i=0;i<count;i++)
 scanf("%d",&num[i]);

 bubble(num,count);

 printf("\nSorted integers\n");
 display(num,count);

 printf("\nEnter integer to be searched: ");
```

```
scanf("%d",&key);

num[i]=key;

for(i=0;i<count+1;i++)
{
 char a[count];
 snprintf(a,sizeof(int),"%d",num[i]);

 argp[i] = malloc(sizeof(a));
 strcpy(argp[i],a);
}

argp[i] = NULL;

p_id = fork();

if(p_id==0)
{
 execve(argv[1],argp,NULL);
 perror("Child process");
 }

 return 0;
}
```

//Binary search program(child)

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(int argc,char* argv[],char* envp[])
{
 int i,j,arr[argc],key,mid,n;

 for(j=0;j<argc-1;j++)
 {
  n=atoi(argv[j]);
  arr[j]=n;
 }

key=atoi(argv[j]);
i=0;
j=argc-1;
mid=(i+j)/2;

while(arr[mid]!=key && i<=j)
{
 if(key>arr[mid])
 i=mid+1;
 else
 j=mid-1;

 mid=(i+j)/2;
}

if(i<=j)
printf("\n%d is present in the array\n", key);
```

```
 else
 printf("\n%d is not present in the array\n", key);

 return 0;
}
```

Output:

```
Enter no. of integers to be sorted: 7

Enter integers
7
6
5
4
3
2
1

Sorted integers
1    2    3    4    5    6    7

Enter integer to be searched: 4

4 is present in the array
```