

child programm

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

void main(int argc,char*argv[])
{
    int c=0,first,last,middle,n,search, array[10];

    FILE*file;
    //printf("file name:%s",argv[1]);
    file= fopen(argv[1],"r");
    printf("\nfile name:%s\n",argv[1]);
    while((fscanf(file,"%d",&array[c]))!=EOF)

    {
        printf("\n%d",array[c]);
        c++;
    }
    n=c;

    printf("\n enter value to find:");
    scanf("%d",&search);

    first=0;
    last=n-1;
    middle=(first+last)/2;

    while(first<=last)
    {
        if(array[middle]< search)
            first=middle+1;
        else if(array[middle]==search)
        {
            printf(" %d found at location%d.\n",search,middle+1);
            break;
        }
        else
            last=middle-1;

        middle= (first+last)/2;
    }
    if(first>last)
        printf("not found!%d is not present in the list.\n",search);
}
```

parrent programm

```
#include<string.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#define MAX 25

//forword quicksort
void quicksort(int qarr[], int,int);

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

{
```

```

FILE* file;
    int pid,status;
    int merge[MAX],i,n,quick[MAX];
    char*newargv[]={NULL,"sort.txt",NULL};

printf("enter the total number of elements:");
scanf("%d",&n);

printf("enter the elements which to be sort:");
    for(i=0;i<n;i++)
    {
        scanf("%d",&merge[i]);
    }
    for(i=0;i<n;i++)
    {
        quick[i]=merge[i];
    }
newargv[0]=argv[1];

pid=fork();
if(pid==0)
{

    execve(argv[1],newargv,NULL);
}
else if(pid>0)
{
    quicksort(quick,0,n-1);
    printf("\nsorted element by parent(quick sort):");

    file= fopen("sort.txt","w");
    if(file==NULL)
    {
        printf("error: can't open file.\n");
    }
else
    {
        printf("file open successfully.\n");
    }
    for(i=0;i<n;i++)
    {

        fprintf(file,"%d ",quick[i]);

    }
    fclose(file);
    wait(&status);
}
else
    printf("\nerror.\n");

}

void quicksort(int x[10],int first,int last){
    int pivot,j,temp,i;

    if(first<last){
        pivot=first;
        i=first;
        j=last;

        while(i<j)
        {

```

```
while(x[i]<=x[pivot]&& i<last)
    i++;
while(x[j]>x[pivot])
    j--;
if(i<j){
    temp=x[i];
    x[i]=x[j];
    x[j]=temp;
}
temp=x[pivot];
x[pivot]=x[j];
x[j]=temp;
quicksort(x,first,j-1);
quicksort(x,j+1,last);
}
```

OUTPUT:-

```
[root@localhost ~]# ls
sandesh45
anaconda-ks.cfg
bitnami-owncloud-6.0.3-0-module-linux-x64-installer.run
bitnami-owncloud-6.0.3-0-module-linux-x64-installer.run.part
Desktop
Documents
Downloads
ml
Music
[root@localhost Documents]# gcc sandesh13.c -
o p [root@localhost Documents]# gcc
sandesh11.c -o c [root@localhost Documents]#
./p ./c
enter the total number of elements:3
enter the elements which to be sort:5
9
6

sorted element by parent(quick sort):file open successfully.

file name:sort.txt

5
6
9
enter value to find:6
6 found at location2.
[root@localhost Documents]# clear
```