NAME: Kaustubh Gajanan Indulkar
TE-IT(A) 35026

ASSIGNMENT 2 (B)

Problem Statement : Implement the C program in which main program accepts an array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an array and passes the sorted array to child process through the command line arguments of EXECVE system call. The child process uses EXECVE system call to load new program which display array in reverse order.

//testa1.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<unistd.h>
#include<errno.h>
#include<sys/wait.h>
#define MAX 50

void main(int argc, char *argv[])
{
  int n;
  printf("Enter the number of elements : ");
  scanf("%d", &n);
  int arr[n];

  pid_t pid;
  int status;
  char *newenviron[]={NULL};
  char *newargv[20];
  char charArray[256];
  char temp[50];

  charArray[0] = '\0';

for(int i=0; i<n; i++)
{
  printf("Enter element : ");
  scanf("%d",&arr[i]);
}

for(int i=0; i<n; i++)
{
sprintf(temp, "%d", arr[i]);
strcat(charArray, temp);
}
```

NAME: Kaustubh Gajanan Indulkar
TE-IT(A) 35026

```
printf(" program 1 value of n %s \n", charArray);

newargv[0] = argv[1];
newargv[1] = charArray;
newargv[2]=NULL;

pid=fork();

if(pid==-1)
printf("Error on fork\n");

if(pid==0)
{
  //child process
  execve(argv[1], newargv, newenviron);
  printf("execve is successfully executed");
}
else if(pid>0)
  {
    printf("Parent process\n");
    wait(&status);
  }
}
```

// test2.c

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

int main(int argc, char *argv[])
{
  int length = strlen(argv[1]);
  printf("Program 2 value of n= \n ");
  for(int i=length - 1; i>=0; i--)
{
  printf("%d ", argv[1][i] - 48);
}
exit(EXIT_SUCCESS);
}
```

NAME: Kaustubh Gajanan Indulkar
TE-IT(A) 35026

OUTPUT: