**ASSIGNMENT 2B**

Name : Gayatri Sakure

Roll no. : 33268

**CODE:**

1. **FILE NAME: 2b1.c**

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/wait.h>

#include<string.h>

void bass(int arr[25], int n)

{

int i, j, temp;

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

{

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

{

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

{

temp = arr[j];

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

arr[j + 1] = temp;

}

}

}

printf("\n Ascending Order \n");

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

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

printf("\nElements Sorted Using Bubble Sort\n");

printf("\n");

}

void forkeg(int arr[25],int n)

{

int i,status;

int pid = fork();

if (pid == 0)

{

sleep(7);

printf("child process\n");

printf("child process id=%d\n", getpid());

int i;

char \*arg[n];

char ch[25];

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

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

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

snprintf(ch,sizeof(int),"%d",arr[i]);

arg[i]=malloc(sizeof(ch));

strcpy(arg[i],ch);

}

arg[i]=NULL;

execve("./2b2",arg,NULL);

printf("child process id=%d\n", getpid());

printf("parent process id=%d\n", getppid());

printf("child process has ended\n");

}

else

{

printf("parent process\n");

printf("parent process id=%d\n", getppid());

bass(arr, n);

printf("Now moving to child process\n");

wait(&status);

printf("Back in parent\n\n\n");

}

}

int main()

{

int arr[25],n;

printf("\nEnter the no of values in array :- ");

scanf("%d", &n);

printf("Enter the array elements : ");

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

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

forkeg(arr,n);

return 0;

}

1. **FILE NAME: 2b2.c**

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/wait.h>

#include<string.h>

#include<stdbool.h>

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

printf("2b2 process id=%d\n", getpid());

int arr[argc],temp,j,i;

printf("\nProgram inside 2b2\n");

for(int i=0;i<argc;i++){

arr[i]=atoi(argv[i]);

}

printf("\nArray before sorting :-");

for(int i=0;i<argc;i++){

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

}

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

for(j = 0; j < argc-i-1 ; ++j){

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

temp = arr[j];

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

arr[j+1] = temp;

}

}

}

printf("\nArray after sorting :-");

for(int i=0;i<argc;i++){

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

}

printf("\nexiting the program\n");

return 0;

}

**OUTPUT:**

VirtualBox:~/33268\_OS$ gcc 2b2.c -o 2b2

VirtualBox:~/33268\_OS$ gcc 2b1.c -o b

VirtualBox:~/33268\_OS$ ./b

VirtualBox:~/33268\_OS$ ./b

Enter the no of values in array :- 5

Enter the array elements : 10 19 25 14 3

parent process

parent process id=3121

Ascending Order

3 10 14 19 25

Elements Sorted Using Bubble Sort

Now moving to child process

child process

child process id=3553

10

19

25

14

3

2b2 process id=3553

Program inside 2b2

Array before sorting :-

10

19

25

14

3

Array after sorting :-

25

19

14

10

3

exiting the program

Back in parent