Assignment 2B

Main

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

void asc(int *a, int sz);

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

 int arr[10];
 int i,size;
 char *env[]={NULL};
 FILE *f;

 char *newarg[]={NULL,"sort.txt", NULL};
 newarg[0]=argv[1];

 pid_t pid;

 printf("\nEnter size of array: ");
 scanf("%d", &size);

 printf("\nEnter %d elements: \n",size);
 for(i=0; i<size; i++)
 {
  scanf("%d", &arr[i]);
 }
```

```
 pid = fork();

 if(pid == 0)
  {//sleep(1);
  execve(argv[1],newarg,env);
  //exit(0);
  }
 else
 {
  asc(arr, size);
  f=fopen("sort.txt","w");

 fprintf(f," %d",size);
 for(i=0;i<size;i++)
 {
 fprintf(f," %d",arr[i]);
 }

  fclose(f);
 }

 return 0;
}

void asc(int *a,int sz)
{
  int i, j;
  int temp;
```

```
  for(i=0; i<sz; i++)
  {
   for(j=i+1; j<sz; j++)
   {
      if(a[i]>a[j])
      {
      temp = a[i];
      a[i] = a[j];
      a[j] = temp;
      }
   }
  }
  printf("\n Parent Process");
  printf("\nSorted array in ascending order is:\n");
  for(i=0; i<sz; i++)
  {
   printf("%d\t", a[i]) ;
  }
  printf("\n");
}
```

Output:

(base) student@student-ThinkCentre-M700:~$ cd Documents (base)
student@student-ThinkCentre-M700:~/Documents$ mkdir ass2 (base)
student@student- ThinkCentre-M700:~/Documents$ gcc ass2a.c lpthread (base)
student@student-ThinkCentre-M700:~/Documents$ ./a.out

Enter size of array: 5

Enter 5 elements:

8

3

5

9

2

Parent Process

Sorted array in ascending order is:

2 3 5 8 9

Assignment2B

Child

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

void desc(int *a, int sz);

int main(int argc,char *argv[])
{
int arr[10],i,size;
FILE *f;

f=fopen(argv[1],"r");

printf(" %s",argv[1]);
fscanf(f,"%d",&size);
printf(" Array Size is: %d\n",size);

for(i=0;i<size;i++)
{
fscanf(f," %d",&arr[i]);

printf(" %d",arr[i]);
}
desc(arr, size);

return(0);
```

```
}

void desc(int *a, int sz)
{
  int i, j;
  int temp;

  for(i=0; i<sz; i++)
  {
   for(j=i+1; j<=sz; j++)
   {
      if(a[i]<a[j])
      {
      temp = a[i];
      a[i] = a[j];
      a[j] = temp;
      }
   }
  }
  printf("\n Child Process");
  printf("\nSorted array in descending order is:\n");
  for(i=0; i<sz; i++)
  {
   printf("%d\t", a[i]) ;
  }
  printf("\n");
}
```

Sort.txt

Array Size is: 5

2 3 5 8 9

Child Process

Sorted array in descending order is:

9 8 5 3 2