Mihir Kulkarni 33132 L9
Program no1. A2(b)

```
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
#include <string.h>
#include <malloc.h>
#include <stdlib.h>
#include <sys/types.h>

void bubble_sort(int arr[],int size)
{
        int temp;
        for(int i=0;i<size-1;i++)
        {
                for(int j=0;j<size-i-1;j++)
                {
                        if(arr[j]>arr[j+1])
                        {

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

        printf("\nSorted elements:-\n");
   for(int i=0;i<size;i++)
        printf("%d ",arr[i] );
}

int main()
{
        int intArray[15],size;
        char buffer[15],*arg[15]; //array of strings
        int pid;
        printf("\nEnter the size of array: ");
        scanf("%d",&size);
        printf("\nEnter the elements in array: ");
        for(int i=0;i<size;i++)
        {
                scanf("%d",&intArray[i]);
        }
        printf("Sorting elements:-\n");
        bubble_sort(intArray,size);
        printf("\n-------------------------------------------------------");
        printf("\nNow invoking fork\n");
        printf("-------------------------------------------------------\n");
        pid=fork();
        if(pid==0)
        {
                //wait(NULL);

                printf("Into the child process\n");
                printf("-------------------------------------------------------\n");
                printf("\nSorted Numbers: ");
                //converting into string format
                for(int i = 0;i < size;i++)
                {
                        sprintf(buffer, "%d", intArray[i]);
                        arg[i] = malloc(sizeof(buffer));
                        strcpy(arg[i], buffer);
                }

                arg[size] = NULL;

                for(int i=0;i<size;i++)
                printf("%s",arg[i]);
                printf("\nInvoking another program\n");
                printf("\nExecuting execve.\n");
                execve("./2b2",arg,NULL);
                printf("\nExecve completed.\n");
                printf("\nChild Process completed. Child exiting.\n");
                printf("-------------------------------------------------------\n");

        }
        else if(pid>0)
        {
                printf("\nIn Parent process. Waiting.\n");
                printf("-------------------------------------------------------\n");
                sleep(10);
                //wait(0);
                printf("\nParent execution complete. Exiting.\n");
        }
        return 0;
}
```

Program no2. A2(b)[binary search]

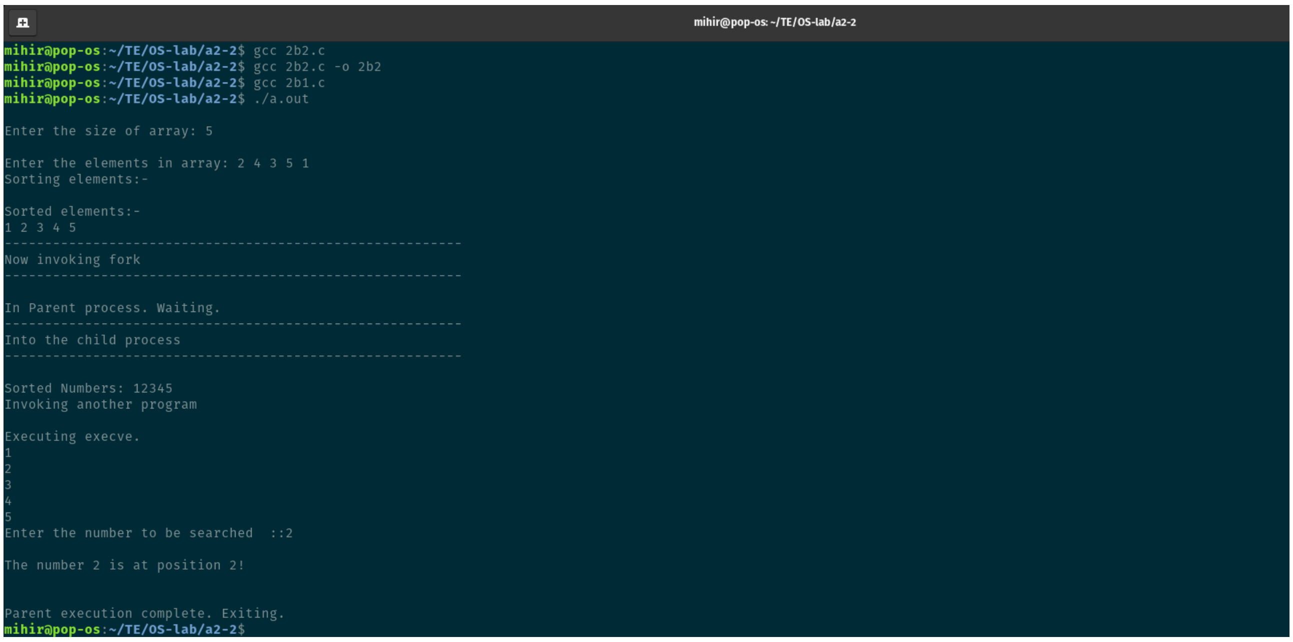
```
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
#include<stdlib.h>
#include<string.h>

int main(int argc,char *argv[])
{
        int arr[15];
        int noOfElements = argc;
        for(int i = 0; i < noOfElements;i++)
        {
                arr[i] = atoi(argv[i]);                 //conversion back to int
                printf("%d\n",arr[i]);
        }

        int low = 0,mid, high = noOfElements-1;

        int temp;
        printf("Enter the number to be searched  ::");
        scanf("%d",&temp);
        while(low <= high)
        {
                mid = (high+low)/2;
                if(arr[mid] == temp)
                {
                        printf("\nThe number %d is at position %d!\n",temp,mid+1);
                        break;
                }
                else if (arr[mid] > temp){
                        high = mid-1;
                        continue;
                }
                else{
                        low = mid+1;
                        continue;
                }
        }
        if(low > high){
                printf("\nNot found");
        }

}
```



```
mihir@pop-os:~/TE/OS-lab/a2-2$ gcc 2b2.c
mihir@pop-os:~/TE/OS-lab/a2-2$ gcc 2b2.c -o 2b2
mihir@pop-os:~/TE/OS-lab/a2-2$ gcc 2b1.c
mihir@pop-os:~/TE/OS-lab/a2-2$ ./a.out

Enter the size of array: 5

Enter the elements in array: 2 4 3 5 1
Sorting elements:-

Sorted elements:-
1 2 3 4 5
---------------------------------------------------------
Now invoking fork
---------------------------------------------------------

In Parent process. Waiting.
---------------------------------------------------------
Into the child process
---------------------------------------------------------

Sorted Numbers: 12345
Invoking another program

Executing execve.
1
2
3
4
5
Enter the number to be searched  ::2

The number 2 is at position 2!


Parent execution complete. Exiting.
mihir@pop-os:~/TE/OS-lab/a2-2$
```