



Graphic Era

HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Data Structures And File Organization End Term File

Submitted To: Mrs Manisha Koranga

Submitted By: Babita Karmiyal



NAME: Babita Karmiyal
COURSE: BCA-HALDWANI

Data Structures And File

Organizations

TBC 201

INDEX

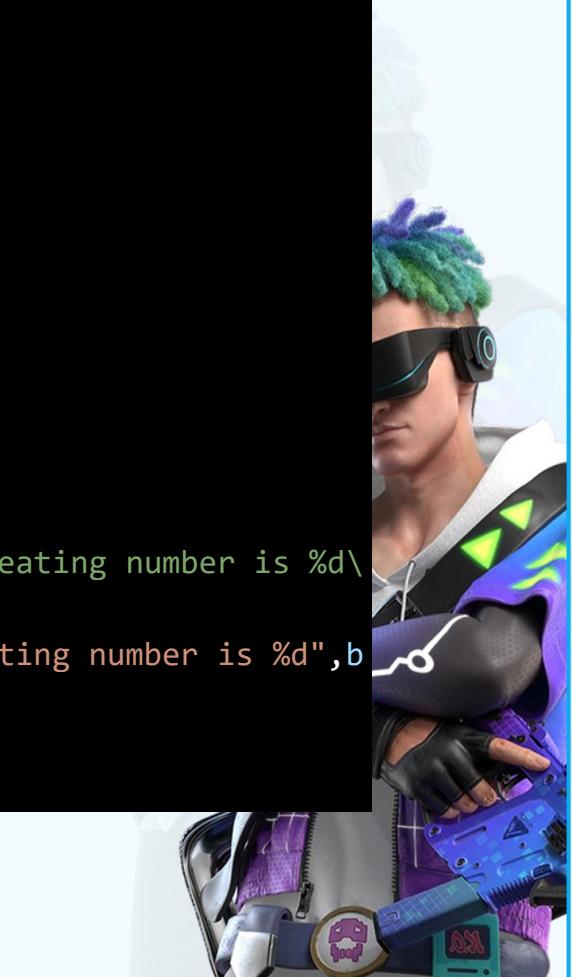
Sr. No.	Assign. Date	Subm. Date	<u>Title</u>	Teacher's Remark
1.		23 APRIL 2021	Lab Assignment 1	
2.		30 APRIL 2021	Lab Assignment 2	
3.		13 MAY 2021	Lab Assignment 3	
4.		13 MAY 2021	Lab Assignment 4	
5.		21 MAY 2021	Lab Assignment 5	
6.		26 MAY 2021	Lab Assignment 6	
7.		27 MAY 2021	Lab Assignment 7	
8.		03 August 2021	Lab Assignment 8	
9.		27 August 2021	Lab Assignment 9	

Lab Assignment 1

01_second_non_repeating_num_in_array

```
#include<stdio.h>
int main()
{
    int a[10],b[2],freq=0, count=0;
    printf("Enter the elements of the array : \n");
    for(int i=0; i<10; i++)
        scanf("%d",&a[i]);

    for(int i=0; i<10; i++)
    {
        freq=1;
        for(int j=0; j<10; j++)
        {
            if(a[i]==a[j]&& i!=j)
            {
                freq++;
            }
        }
        if(freq==1)
        {
            b[count]=a[i];
            count++;
        }
        if(count==2)break;
    }
    // printf("\n\nThe first non repeating number is %d\n",b[0]);
    printf("\n\nThe second non repeating number is %d",b[1]);
    return 0;
}
```



```

File Edit Selection View Go Run Terminal Help 01_second_non_repeating_num_in_array.cpp - DS-ALGO - Visual Studio Code
and_small_in_Array.cpp 05_occurrence_of_any_num_in_Array.cpp 01_multiplicationOfMatrices.cpp 01_second_non_repeating_num_in_array.cpp > ...
EXPLORER ... DS labAssignment1 > 01_second_non_repeating_num_in_array.cpp > main()
... Welcome
... 01_binarySearch.cpp oth...
... 02_reverse_Array_using_...
... 03_common_element_of...
... 04_interchange_large_and...
... 05_occurrence_of_any_num...
... 01_multiplicationOfMatr...
... 01_second_non_repeating...
... DS-ALGO
... > 03DS_Assignment
... > DS labAssignment1
... > .vscode
... > 01_flowchart.pdf
... > 01_second_non_repeating...
... > 01_second_non_repeating...
... > 02_flowchart.pdf
... > 02_reverse_Array_using_swa...
... > 02_reverse_Array_using_swa...
... > 03_flowchart.pdf
... > 03_common_element_of_an...
... > 03_common_element_of_an...
... > 04_flowchart.pdf
... > 04_interchange_large_and_sm...
... > 04_interchange_large_and_sm...
... > 05_flowchart.pdf
... > 05_occurrence_of_any_num_i...
... > 05_occurrence_of_any_num_i...
... > OUTLINE
... > 0 0 △ 0 Live Share
Windows Search the web and Windows PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE Enter the elements of the array : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 The second non repeating number is 3 PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\DS labAssignment1 > []
Ln 24, Col 21 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 12:58 PM 01-Sep-21

```

02_reverse_Array_using_swapping

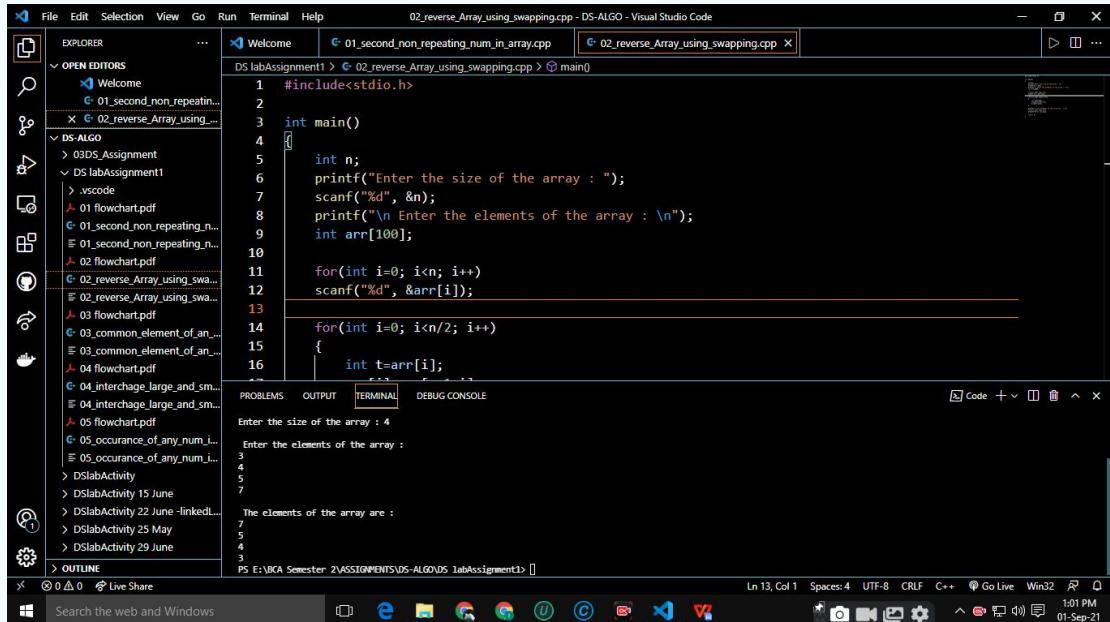
```

#include<stdio.h>

int main()
{
    int n;
    printf("Enter the size of the array : ");
    scanf("%d", &n);
    printf("\n Enter the elements of the array : \n");
    int arr[100];
    for(int i=0; i<n; i++)
        scanf("%d", &arr[i]);
    for(int i=0; i<n/2; i++)
    {
        int t=arr[i];
        arr[i]=arr[n-1-i];
        arr[n-1-i]=t;
    }
    printf("\n The elements of the array are : \n");
    for(int i=0; i<n; i++)
        printf("%d\n", arr[i]);
}

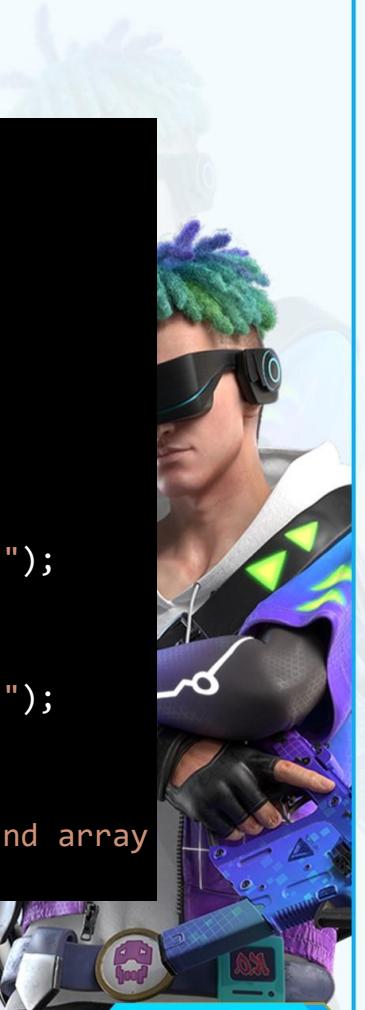
```

```
        return 0;  
}
```



```
#include<stdio.h>  
  
int main()  
{  
    int n;  
    int arr1[100], arr2[100];  
    printf("Enter the size of the array : ");  
    scanf("%d", &n);  
  
    printf("\n Enter the elements of the array1 : \n");  
    for(int i=0; i<n; i++)  
        scanf("%d", &arr1[i]);  
    printf("\n Enter the elements of the array2 : \n");  
    for(int i=0; i<n; i++)  
        scanf("%d", &arr2[i]);  
    printf("\n \nThe common elements of the array1 and array  
2 are : \n");
```

03_common_element_of_an_Array



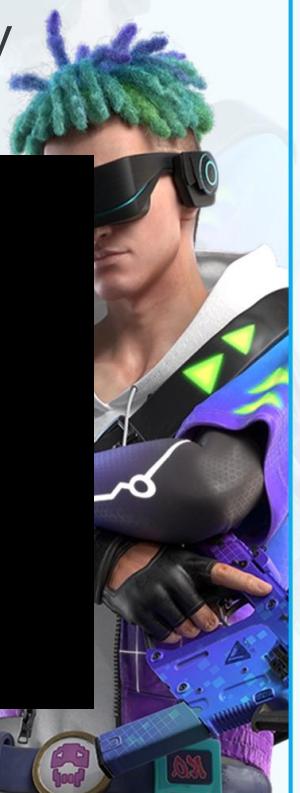
```
#include<stdio.h>  
  
int main()  
{  
    int n;  
    int arr1[100], arr2[100];  
    printf("Enter the size of the array : ");  
    scanf("%d", &n);  
  
    printf("\n Enter the elements of the array1 : \n");  
    for(int i=0; i<n; i++)  
        scanf("%d", &arr1[i]);  
    printf("\n Enter the elements of the array2 : \n");  
    for(int i=0; i<n; i++)  
        scanf("%d", &arr2[i]);  
    printf("\n \nThe common elements of the array1 and array  
2 are : \n");
```

```
for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
        if(arr1[i]==arr2[j])
            printf("%d\n", arr1[i]);
}
return 0;
}
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists "OPEN EDITORS" and "DS ALGO" sections. The "DS ALGO" section contains files like "03DS_Assignment", "DS labAssignment1", "vscode", "flowchart.pdf", and several C files: "01_second_non_repeating_num_in_array.cpp", "02_reverse_Array_using_swapping.cpp", and "03_common_element_of_an_Array.cpp".
- Code Editor:** The main area displays C code for finding common elements between two arrays. The code uses `#include <stdio.h>` and defines `main()`. It prompts for array sizes and elements, then prints common elements.
- Terminal:** Below the code editor, the terminal tab is active, showing the command "PS E:\BCA Semester 2\ASSIGNMENTS\DS ALGO\DS labAssignment1" and the output "The common elements of the array1 and array2 are :".
- Status Bar:** At the bottom, it shows "Ln 11, Col 27", "Spaces: 4", "UTF-8", "C++", "Go Live", and "Win32".

04_interchage_large_and_small_in_Array



```
#include<stdio.h>

int main()
{
    int n;
    int arr1[100];
    printf("Enter the size of the array : ");
    scanf("%d", &n);

    printf("\n Enter the elements of the array : \n");
    for(int i=0; i<n; i++)
        scanf("%d", &arr1[i]);
}
```

```
printf("\n\n\n The elements of the array : \n");
for(int i=0; i<n; i++)
printf("%d\n", arr1[i]);
```

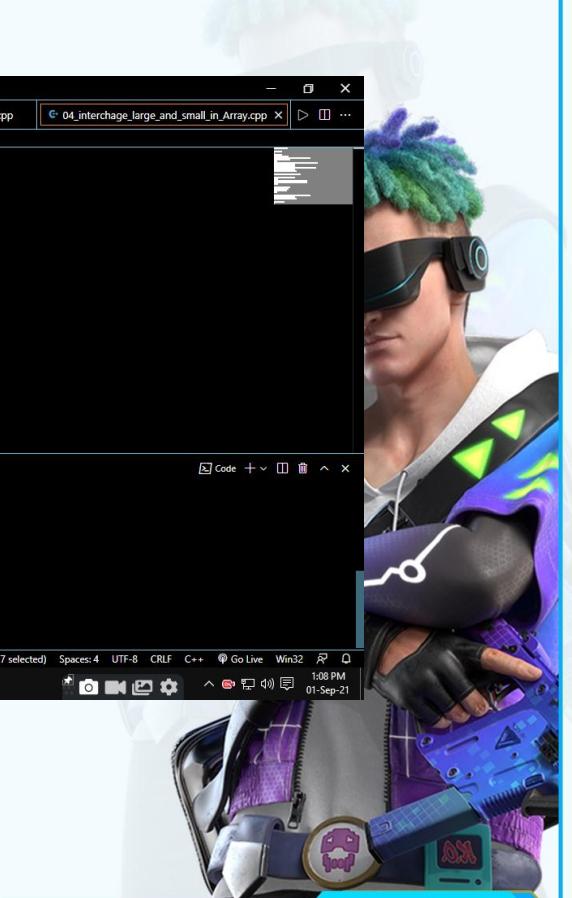
```
int *small=arr1, *large=arr1;
```

```
for(int i=0; i<n; i++)
{
    if(arr1[i]<*small)small=&arr1[i];
    if(arr1[i]>*large)large=&arr1[i];
}
```

```
int temp=*small;
*small=*large;
*large=temp;
```

```
printf("\n\nAfter Interchanging !!!!\n");
for(int i=0; i<n; i++)
printf("%d\n", arr1[i]);
```

```
return 0;
}
```



A screenshot of Visual Studio Code showing the C++ code for swapping the largest and smallest elements in an array. The code includes prompts for array size and elements, and prints the results before and after the swap. The interface shows multiple tabs open, including files for other assignments. The terminal at the bottom shows the execution of the program.

```
#include<stdio.h>
int main()
{
    int n;
    int arr1[100];
    printf("Enter the size of the array : ");
    scanf("%d", &n);

    printf("\n Enter the elements of the array : \n");
    for(int i=0; i<n; i++)
        scanf("%d", &arr1[i]);
    printf("\n\n\n The elements of the array : \n");
    for(int i=0; i<n; i++)
        printf("%d\n", arr1[i]);

    int *small=arr1, *large=arr1;
    for(int i=0; i<n; i++)
    {
        if(arr1[i]<*small)small=&arr1[i];
        if(arr1[i]>*large)large=&arr1[i];
    }

    int temp=*small;
    *small=*large;
    *large=temp;

    printf("\n\nAfter Interchanging !!!!\n");
    for(int i=0; i<n; i++)
        printf("%d\n", arr1[i]);
}

return 0;
}
```

The elements of the array :
21
3
5
3
After Interchanging !!!
3
21
5
3

PS E:\VBCA Semester 2\ASSIGNMENTS\DS-ALGO\DS labAssignment1> []

Ln 1, Col 1 (727 selected) Spaces: 4 UTF-8 CRLF C++ Go Live Win32 1:08 PM 01-Sep-21

05_occurrence_of_any_num_in_Array

```
#include<stdio.h>
int main()
{
    int n;
    int arr1[100];
    printf("Enter the size of the array : ");
    scanf("%d", &n);
    printf("\n Enter the elements of the array : \n");
    for(int i=0; i<n; i++)
        scanf("%d", &arr1[i]);
    printf("\n\n\n The elements of the array : \n");
    for(int i=0; i<n; i++)
        printf("%d\n", arr1[i]);
    int x, count=0;
    printf("Enter the number whose occurrence you wanna find : ");
    scanf("%d",&x);
    for(int i=0; i<n; i++)
        if(arr1[i]==x)
            count++;
    printf("The number %d occurs %d times in the array !!!!", x, count);
    return 0;
}
```





A screenshot of Visual Studio Code showing a C++ file named "05_occurrence_of_any_num_in_Array.cpp". The code implements a function to find the occurrence of a specific number in an array. The user inputs the size of the array and its elements, then specifies the number to search for. The output shows that the number 7 occurs 1 time in the array.

```
#include<stdio.h>
int main()
{
    int n;
    int arr1[100];
    printf("Enter the size of the array : ");
    scanf("%d", &n);
    printf("\nEnter the elements of the array : \n");
    for(int i=0; i<n; i++)
        scanf("%d", &arr1[i]);
    printf("\n\nThe elements of the array : \n");
    for(int i=0; i<n; i++)
        printf("%d\n", arr1[i]);
    int x, count=0;
    printf("Enter the number whose occurrence you wanna find : ");
    scanf("%d", &x);
    for(int i=0; i<n; i++)
        if(arr1[i]==x)
            count++;
    printf("The number %d occurs %d times in the array !!!\n", x, count);
}
```

Lab Assignment 2

01_multiplicationOfMatrices

```
#include<stdio.h>
void mat_elem(int a[10][10], int r, int c)
{
    for(int i =0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Element [%d][%d] : ", i+1, j+1);
            scanf("%d", &a[i][j]);
        }
    }
}
```

```
void show_elem(int a[10][10], int r, int c)
{
    for(int i =0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d\t ", a[i][j]);
        }
        printf("\n");
    }
}
```

```
int main()
{
```

```
int a1[10][10], r1, c1;
int a2[10][10], r2, c2;
int a3[10][10];
```

```
printf("Enter the number of elements in the row & column
of matrix 1 : \n");
scanf("%d",&r1);
scanf("%d", &c1);
```

```
printf("Enter the number of elements in the row & column
of matrix 2 : \n");
scanf("%d",&r2);
scanf("%d", &c2);
```

```
if(c1!=r2)
{
    printf("CAN'T MULTIPLY !!!!!!!\n");
    return 1;
}
```

```
\n\nEnter the elements of matrix 1 : \n");
mat_elem (a1 ,r1,c1);
printf("\n\n\nThe matrix is : \n\n");
show_elem(a1,r1,c1);
```

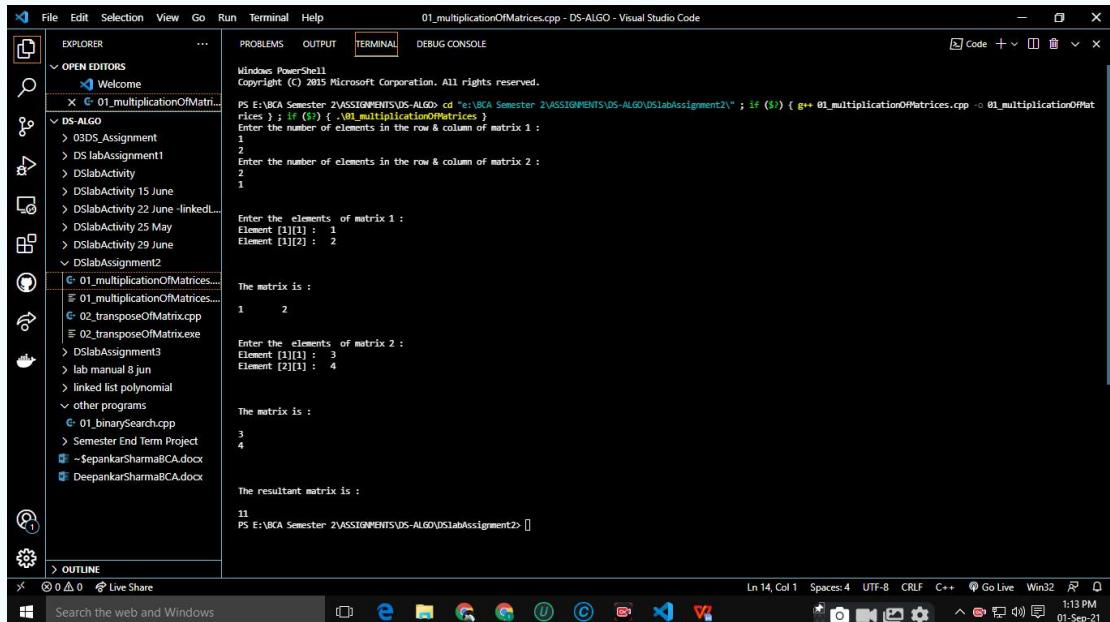
```
\n\nEnter the elements of matrix 2 : \n");
mat_elem(a2,r2,c2);
printf("\n\n\nThe matrix is : \n\n");
show_elem(a2,r2,c2);
```

```
for(int i=0; i<r1; i++)
{
    for(int j=0; j<c2; j++)
    {
        a3[i][j]=0;
        for(int k=0; k<c1;k++ )
            a3[i][j]+=a1[i][k]*a2[k][j];
    }
}
```

```
printf("\n\n\nThe resultant matrix is : \n\n");
```



```
    show_elem(a3,r1,c2);
}
```



```
File Edit Selection View Go Run Terminal Help 01_multiplicationOfMatrices.cpp - DS-ALGO - Visual Studio Code
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment2"
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment2> & .\01_multiplicationOfMatrices
Enter the number of elements in the row & column of matrix 1 :
1
2
Enter the number of elements in the row & column of matrix 2 :
2
2
1

Enter the elements of matrix 1 :
Element [1][1] : 1
Element [1][2] : 2

The matrix is :
1     2

Enter the elements of matrix 2 :
Element [1][1] : 3
Element [2][1] : 4

The matrix is :
3
4

The resultant matrix is :
11
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment2>
```

02 transposeOfMatrix

```
#include<stdio.h>
void mat_elem(int a[10][10], int r, int c)
{
    for(int i =0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("Element [%d][%d] : ", i+1, j+1);
            scanf("%d", &a[i][j]);
        }
    }
}
```

```
void show_elem(int a[10][10], int r, int c)
{
    for(int i =0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d\t ", a[i][j]);
        }
        printf("\n");
    }
}
```

```
void tps_mat(int a[10][10], int r, int c)
{
    printf("\n\nThe transpose of the matrix is : \n");
    for(int i =0; i<r; i++)
    {
        for(int j=0; j<c; j++)
        {
            printf("%d\t ", a[j][i]);
        }
        printf("\n");
    }
}
```

```
int main()
{
    int a1[10][10], r1, c1;
    // int a2[10][10], r2, c2;
    // int a3[10][10];
```

```
    printf("Enter the number of elements in the row & column
of matrix 1 : \n");
    scanf("%d",&r1);
    scanf("%d", &c1);
```

```

    // printf("Enter the number of elements in the row & col
umn of matrix 2 : \n");
    // scanf("%d",&r2);
    // scanf("%d", &c2);

    // if(c1!=r2)
    // {
    //     printf("CAN'T MULTIPLY !!!!!!!\n");
    //     return 1;
    // }

printf("\n\nEnter the elements of matrix 1 : \n");
mat_elem (a1 ,r1,c1);
printf("\n\n\nThe matrix is : \n\n");
show_elem(a1,r1,c1);
tps_mat(a1,r1,c1);

// printf("\n\nEnter the elements of matrix 2 : \n");
// mat_elem(a2,r2,c2);
// printf("\n\n\nThe matrix is : \n\n");
// show_elem(a2,r2,c2);

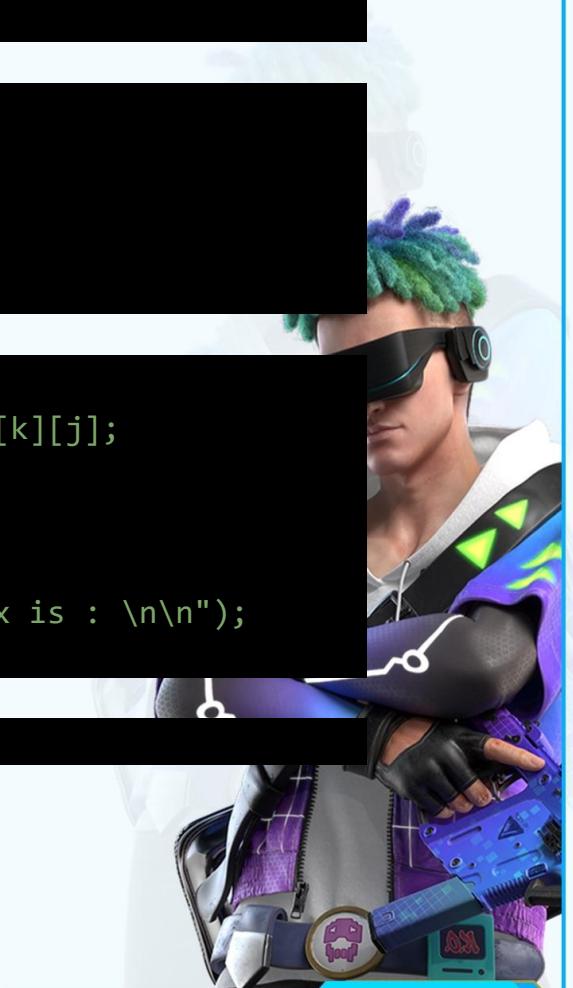
// for(int i=0; i<r1; i++)
// {
//     for(int j=0; j<c2; j++)
//     {
//         a3[i][j]=0;

//         for(int k=0; k<c1;k++ )
//             a3[i][j]+=a1[i][k]*a2[k][j];
//     }
// }

// printf("\n\n\nThe resultant matrix is : \n\n");
// show_elem(a3,r1,c2);

}

```



```

File Edit Selection View Go Run Terminal Help
EXPLORER ... Welcome 01_multiplicationOfMatrices.cpp 02_transposeOfMatrix.cpp
OPEN EDITORS Welcome
C 01_multiplicationOfMatrices.cpp
C 02_transposeOfMatrix.cpp
DS ALGO
> 03DS_Assignment
> DS labAssignment1
> DSlabActivity
> DSlabActivity 15 June
> DSlabActivity 22 June -linked...
> DSlabActivity 25 May
> DSlabActivity 29 June
DSlabAssignment2
C 01_multiplicationOfMatrices...
E 01_multiplicationOfMatrices...
C 02_transposeOfMatrix.cpp
E 02_transposeOfMatrix.exe
DSlabAssignment3
> lab manual 8 jun
> linked list polynomial
other programs
C 01_binarySearch.cpp
> Semester End Term Project
DeepankarSharmaBCA.docx
DeepankarSharmaBCA.docx
OUTLINE
Live Share
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
The matrix is :
1 2
3 4
The transpose of the matrix is :
1 3
2 4
1703936 194272
PS E:\VICA Semester 2\ASSIGNMENTS\DS ALGO\DSlabAssignment2>

```

Lab Assignment 3

01_fibonacci_using_Recursion

```

#include<stdio.h>

int fiboRec(int n)
{
    if(n==0) return 0;
    if(n==1) return 1;
    return(fiboRec(n-1)+fiboRec(n-2));
}

int main()
{
    int n;
    printf("Enter the number : ");
    scanf("%d", &n);
    printf("The series upto %d is: \n", n);
}

```

```

for(int i=0; i<=n; i++)
{
    int z = fiboRec(i);
    printf("%d\t",z );
}
return 0;
}

```

File Edit Selection View Go Run Terminal Help

01_fibonacci.cpp - DS-ALGO - Visual Studio Code

EXPLORER OPEN EDITORS DSLabAssignment3 > 01_fibonacci.cpp > main()

1 #include<stdio.h>

2

3 int fiboRec(int n)

4 {

5 if(n==0) return 0;

6 if(n==1) return 1;

7 return(fiboRec(n-1)+fiboRec(n-2));

8 }

9

10 int main()

11 {

12 int n;

13 printf("Enter the number : ");

14 scanf("%d", &n);

15 printf("The series upto %d is: \n", n);

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

17 printf("%d\t", fiboRec(i));

18 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell Copyright (C) 2015 Microsoft Corporation. All rights reserved.

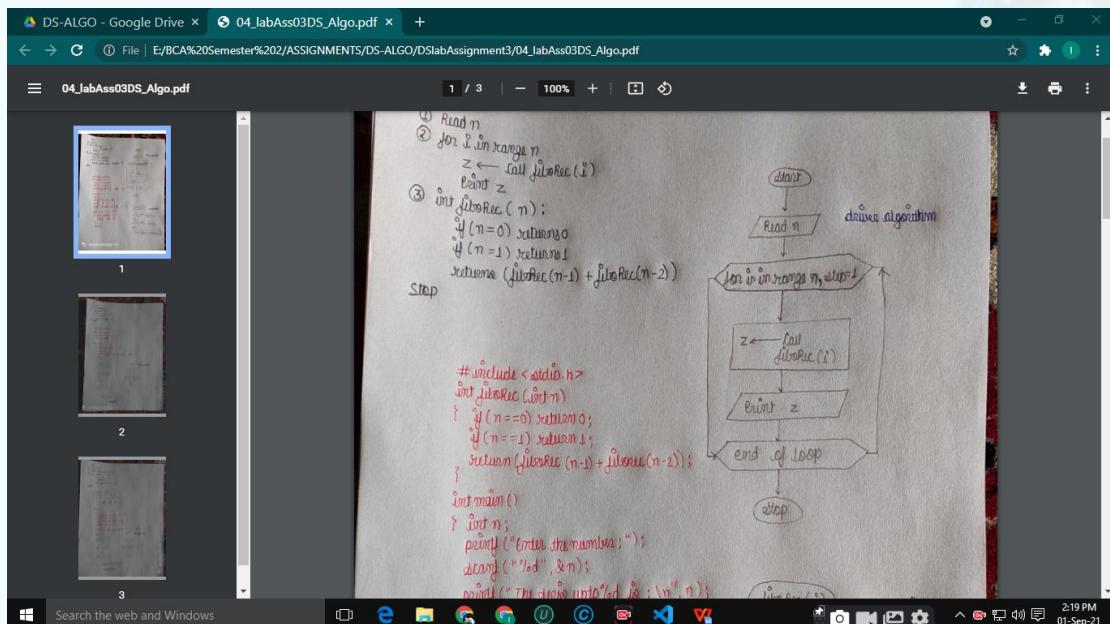
PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment3">; g++ 01_fibonacci.cpp -o 01_fibonacci ; if (\$?) { .\01_fibonacci }

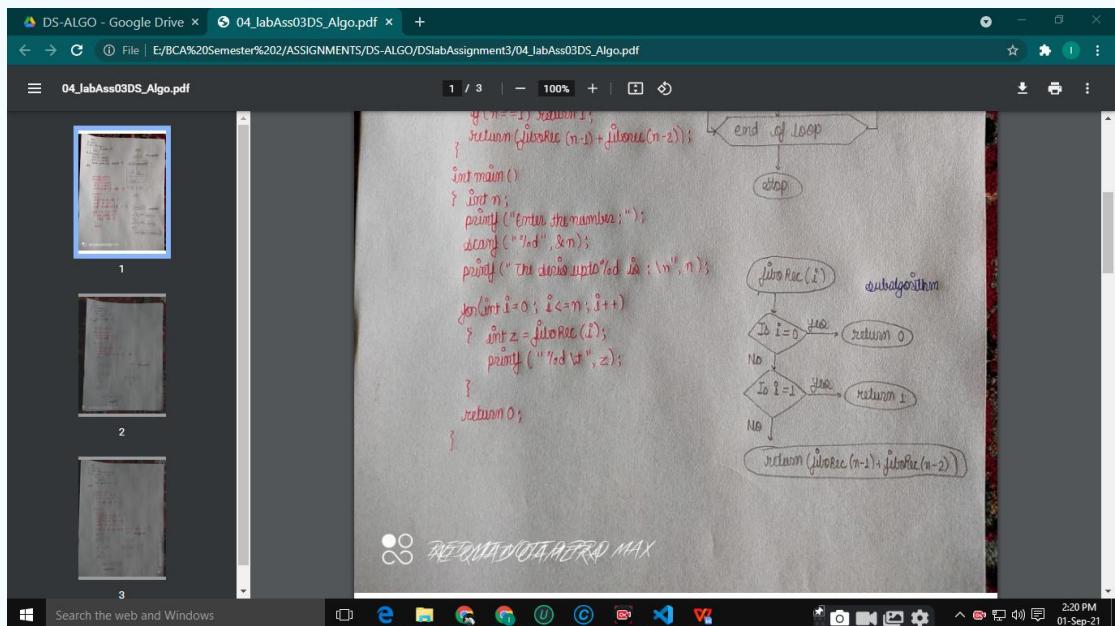
Enter the number : 4

The series upto 4 is:

0 1 1 2 3 PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment3>

Ln 22, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 14:2 PM 01-Sep-21





02_factRec

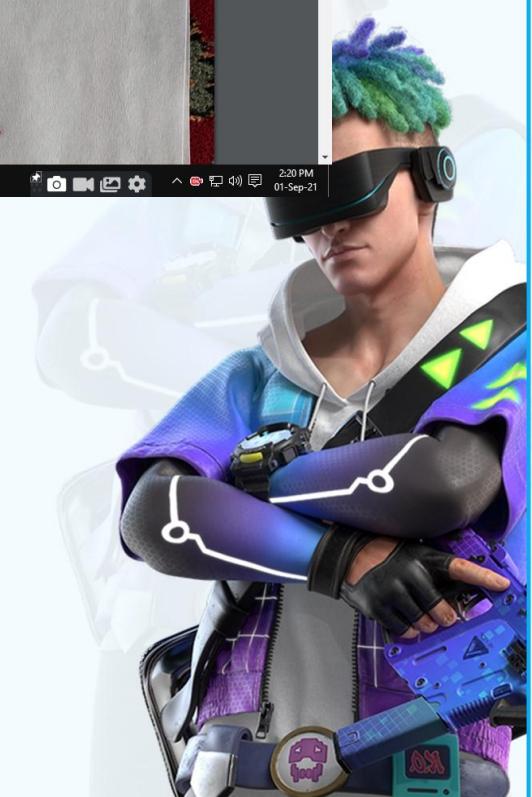
```
#include<stdio.h>

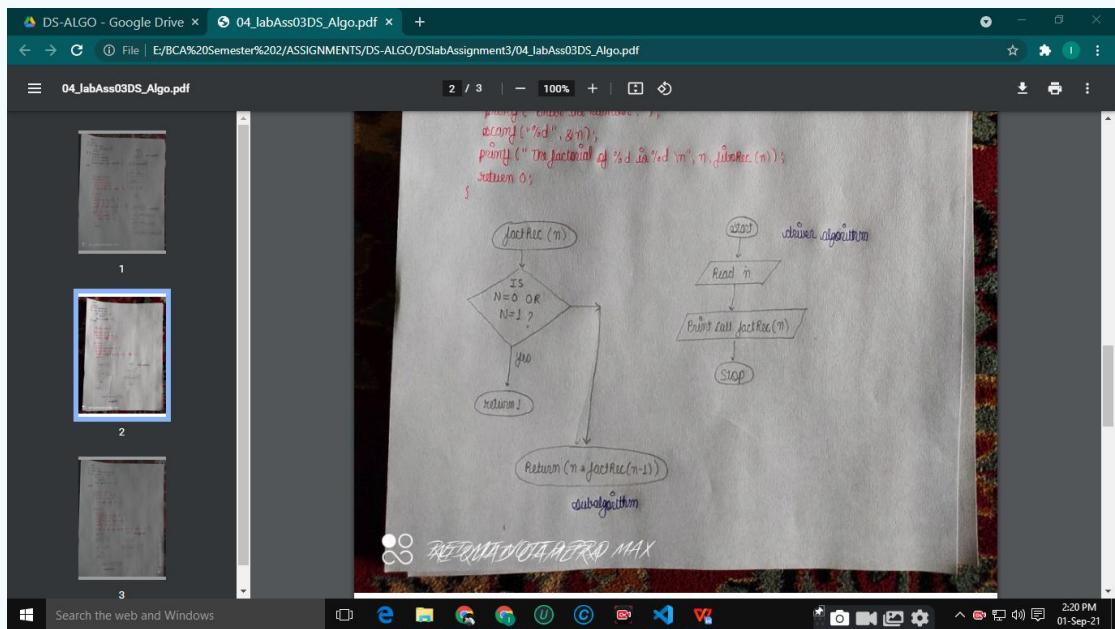
int fiboRec(int n)
{
    if(n==0 || n==1) return 1;
    // if(n==1) return 1;
    return(n*fiboRec(n-1));
}

int main()
{
    int n;
    printf("Enter the number : ");
    scanf("%d", &n);
    printf("The factorial of %d is: %d \n", n,fiboRec(n));
    // for(int i=0; i<=n; i++)
    // {
    //     int z = fiboRec(i);
    //     printf("%d\t",z );
    // }
    return 0;
}
```

Visual Studio Code Terminal Output:

```
PS E:\BCA Semester 2\ASSIGNMENTS\DS_ALGO> cd "e:\BCA Semester 2\ASSIGNMENTS\DS_ALGO\DSlabAssignment3"> if ($_) { g++ 01_fibRec.cpp -o 01_fibRec } ; if ($_) { ./01_fibRec }
Enter the number : 4
The series upto 4 is:
0 1 2 3 PS E:\BCA Semester 2\ASSIGNMENTS\DS_ALGO\DSlabAssignment3> cd "e:\BCA Semester 2\ASSIGNMENTS\DS_ALGO\DSlabAssignment3"> if ($_) { g++ 02_FactRec.cpp -o 02_factRec } ; if ($_) { ./02_factRec }
Enter the number : 4
The factorial of 4 is: 24
PS E:\BCA Semester 2\ASSIGNMENTS\DS_ALGO\DSlabAssignment3>
```





03_gcdRec

```
#include<stdio.h>

int fiboRec(int n, int m)
{
    if(n==0) return m;
    // if(n==1) return 1;
    return fiboRec(m - n, n);
}
```

```
int main()
{
    int n1,n2;
    printf("Enter the first number : ");
    scanf("%d", &n1);
    printf("Enter the second number : ");
    scanf("%d", &n2);
    printf("The Greatest Common Divisor of the two number is
%d", fiboRec(n1,n2));
    // for(int i=0; i<=n; i++)
    // {
    //     int z = fiboRec(i);
    //     printf("%d\t",z );
}
```

```

    // }
    return 0;
}

```

```

File Edit Selection View Go Run Terminal Help
03_gcdRec.cpp - DS-ALGO - Visual Studio Code
EXPLORER OPEN EDITORS DSLabAssignment3 > 03_gcdRec.cpp > hcf(int, int)
Welcome 01_multiplicationOfMatrices.cpp 02_transposeOfMatrix.cpp 01_fibonacciRecursion.cpp 02_factorialRecursion.cpp 03_gcdRec.cpp
OPEN EDITORS DS-ALGO
1 #include <stdio.h>
2 int hcf(int n1, int n2);
3 int main()
4 {
5     int n1, n2;
6     printf("Enter two positive integers: ");
7     scanf("%d %d", &n1, &n2);
8     printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
9     return 0;
10 }
11
12 int hcf(int n1, int n2)
13 {
14     if (n2 != 0)
15     {
16         return hcf(n2, n1 % n2);
17     }
18     else
19     {
20         return n1;
21     }
22 }
23
24 int gcdRec(int n1, int n2)
25 {
26     if (n2 == 0)
27     {
28         return n1;
29     }
30     else
31     {
32         return gcdRec(n2, n1 % n2);
33     }
34 }
35
36 int main()
37 {
38     int n1, n2;
39     printf("Enter the first number: ");
40     scanf("%d", &n1);
41     printf("Enter the second number: ");
42     scanf("%d", &n2);
43     printf("G.C.D of %d and %d is %d.", n1, n2, gcdRec(n1, n2));
44     return 0;
45 }

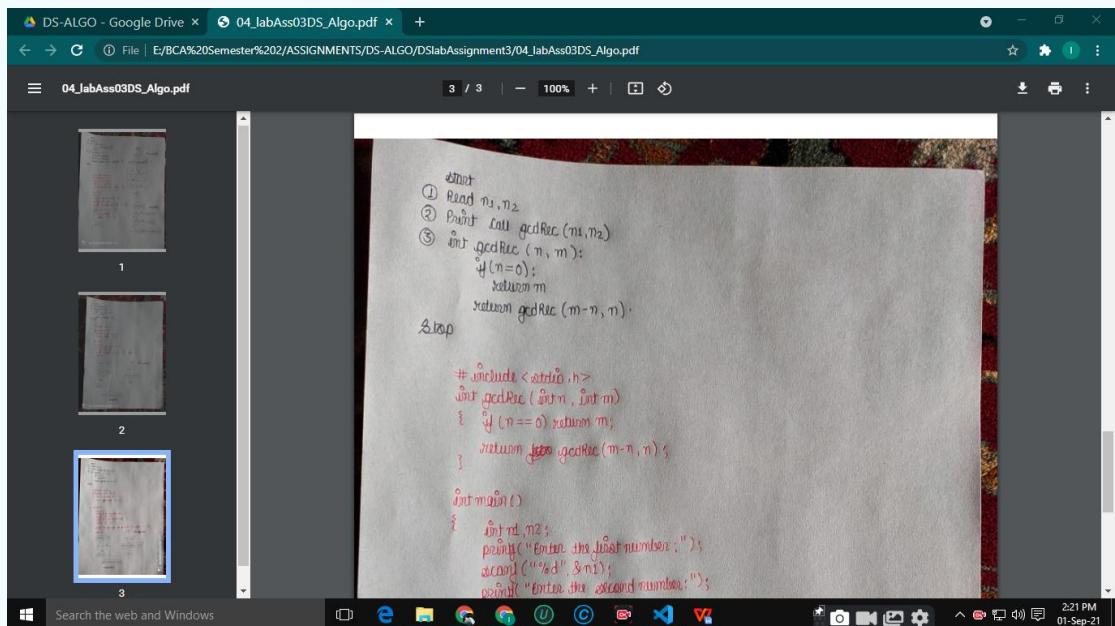
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment3">; g++ 03_gcdRec.cpp -o 03_gcdRec ; ./03_gcdRec
Enter two positive integers: 4 5
G.C.D of 4 and 5 is 1>PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabAssignment3>

Ln 18, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

01-Sep-21



04_labAss03DS_Algo.pdf

```
#include <iostream>
using namespace std;

int gcd(int n1, int n2)
{
    cout << "Enter the first number: ";
    cin >> n1;
    cout << "Enter the second number: ";
    cin >> n2;
    if (n1 < n2) swap(n1, n2);
    cout << "The gcd of the two numbers is " << gcd(n1, n2) << endl;
    return 0;
}

int gcd(int n, int m)
{
    if (n == 0)
        return m;
    else
        return gcd(m % n, n);
}
```

REMOVED OTHER METHODS

Lab Assignment 4

01_ArrayInsertion_StaticArray

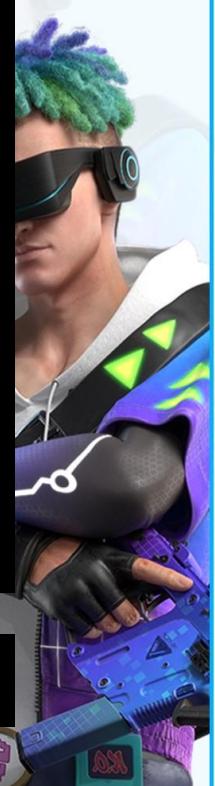
```
#include<stdio.h>

void show(int arr[], int n){

    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int ins(int *La, int n, int item, int k)
{
    if(n>=100)
    {
        printf("The array is full Already !!!!!");
        return -1;
    }
    int j =n-1;
    while(j>=k)
    {
        La[j+1] = La[j];
        j = j-1;
    }
    La[k] = item;
    return 1;
}

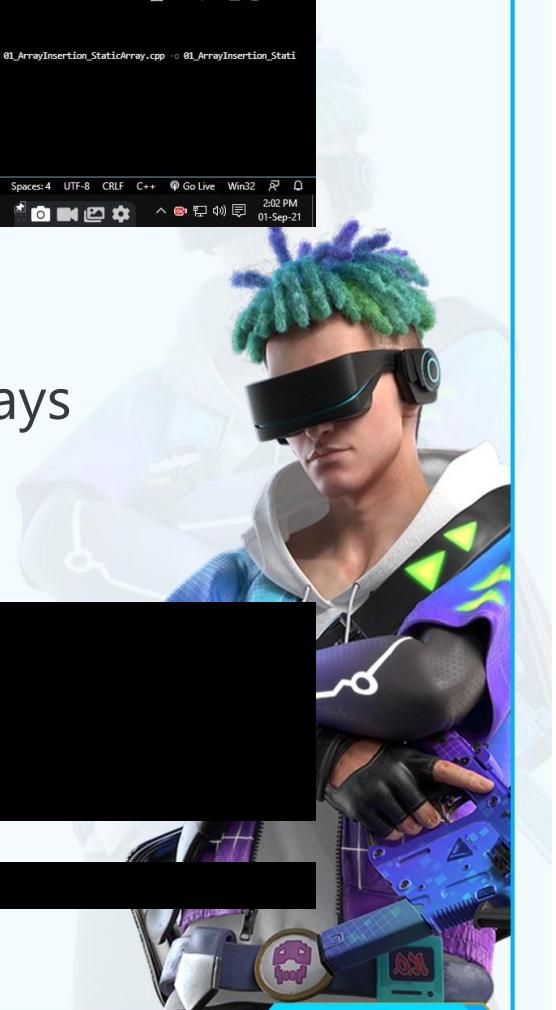
int main()
{
```



```

        int arr[100] = {7, 8, 12, 27, 88};
        int size = 5, element = 45, index=1;
        show(arr, size);
        ins(arr, size, element, index);
        size +=1;
        show(arr, size);
        return 0;
    }
}

```



A screenshot of Visual Studio Code showing a C++ file named `01_ArrayInsertion_StaticArray.cpp`. The code implements an insertion operation into a static array. The terminal below shows the command-line output of the program running.

```

File Edit Selection View Go Run Terminal Help 01_ArrayInsertion_StaticArray.cpp - DS-ALGO - Visual Studio Code
OPEN EDITORS
EXPLORER ... DMatrices.cpp 02_transposeOfMatrix.cpp 01_fibRec.cpp 02_factRec.cpp 03_gcdRec.cpp 01_ArrayInsertion_StaticArray.cpp
DSlabActivity > 01_ArrayInsertion_StaticArray.cpp > show(int[], int)
8     printf("%d ", arr[i]);
9 }
10    printf("\n");
11 }
12
13 int ins(int *La, int n, int item, int k)
14 {
15     if(n>100)
16     {
17         printf("The array is full Already !!!!!");
18         return -1;
19     }
20     int j = n-1;
21     while(j>k)
22     {
23         La[j+1] = La[j];
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity\" ; if ($_) { g++ 01_ArrayInsertion_StaticArray.cpp } & .\01_ArrayInsertion_StaticArray
cArray ) ; IF ($?) { \01_ArrayInsertion_StaticArray
7 8 12 27 88
7 45 8 12 27 88
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity\ []

```

02_ArrayInsertion_DynamicArrays

```

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

void show(int *arr, int n){

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

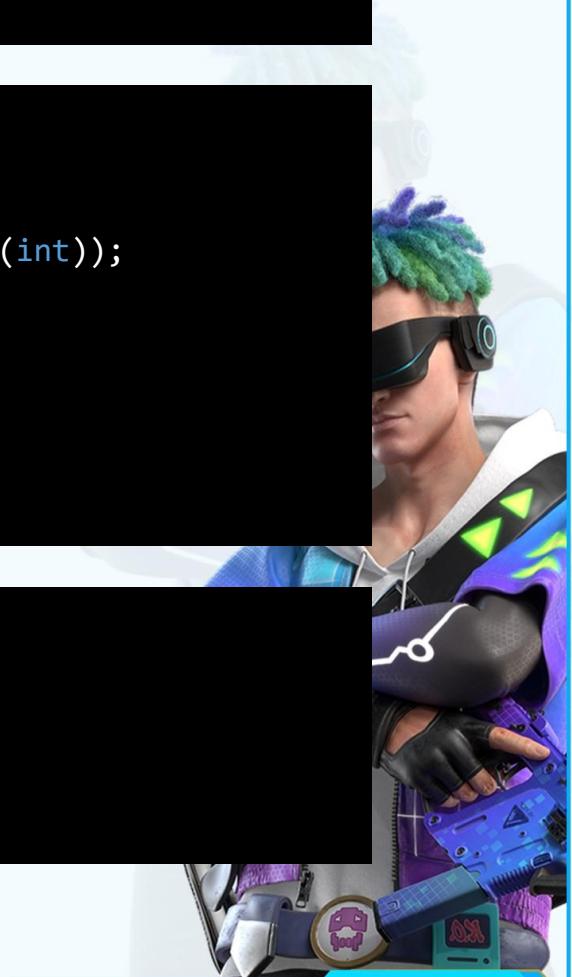
```

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

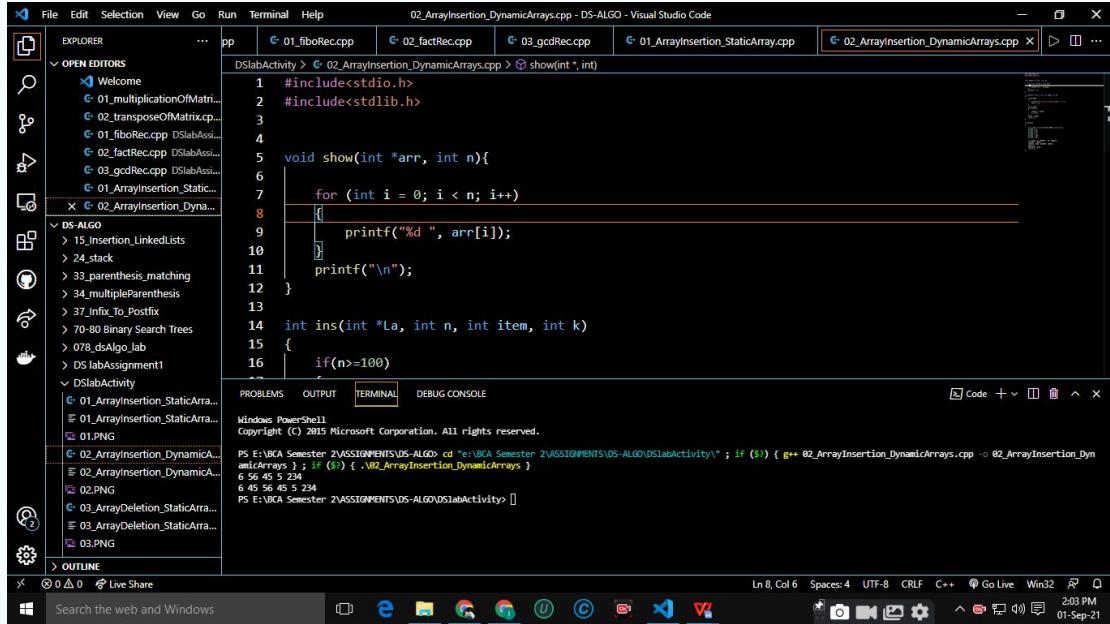
```
int ins(int *La, int n, int item, int k)
{
    if(n>=100)
    {
        printf("The array is full Already !!!!!");
        return -1;
    }
    int j =n-1;
    while(j>=k)
    {
        La[j+1] = La[j];
        j = j-1;
    }
    La[k] = item;
    return 1;
}
```

```
int main()
{
    int * arr = (int*) malloc(100*sizeof(int));
    arr[0] = 6;
    arr[1] = 56;
    arr[2] = 45;
    arr[3] = 5;
    arr[4] = 234;
    arr[5] = 55;
```

```
    int size = 5, element = 45, index=1;
    show(arr, size);
    ins(arr, size, element, index);
    size +=1;
    show(arr, size);
    return 0;
```



}



```
#include<stdio.h>

void show(int *arr, int n){

    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int ins(int *La, int n, int item, int k)
{
    if(n>=100)
        return;
    else
    {
        La[k] = item;
        ins(La, n+1, item, k+1);
    }
}
```

Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity\" ; g++ 02_ArrayInsertion_DynamicArrays.cpp ; .\02_ArrayInsertion_DynamicArrays
6 56 45 5 234
6 45 56 45 5 234
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity>

03_ArrayDeletion_StaticArrays

```
#include<stdio.h>

void show(int arr[], int n){

    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

void delt(int *La, int n, int k)
{
    for (int i = k; i < n-1; i++)
    {
```

```

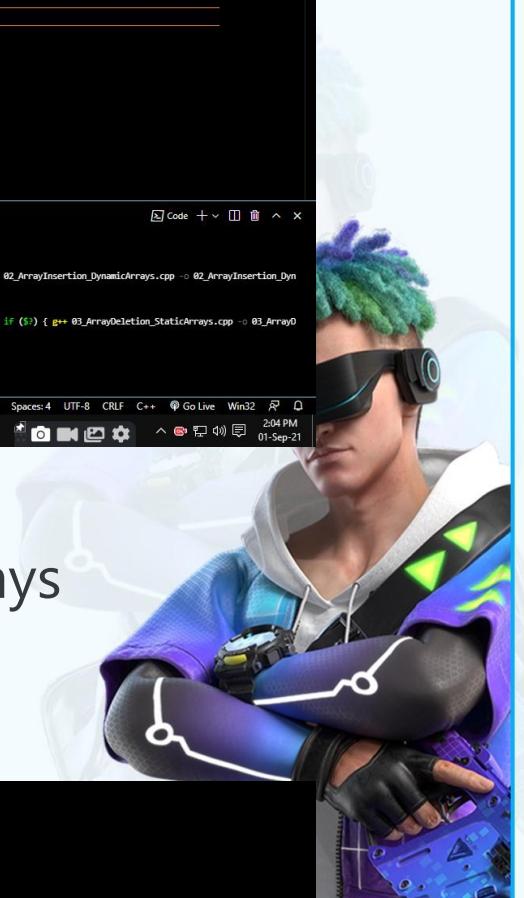
        La[i] = La[i + 1];
    }
}

```

```

int main()
{
    int arr[100] = {7, 8, 12, 27, 88};
    int size = 5, index = 0;
    show(arr, size);
    delt(arr, size, index);
    size -= 1;
    show(arr, size);
    return 0;
}

```



A screenshot of the Visual Studio Code interface showing the code for "03_ArrayDeletion_StaticArrays.cpp". The code implements a deletion operation on a static array. The terminal window shows the command-line interface used to run the program.

```

File Edit Selection View Go Run Terminal Help 03_ArrayDeletion_StaticArrays.cpp - DS-ALGO - Visual Studio Code
EXPLORER OPEN EDITORS 01_gcdRec.cpp 03_gcdRec.cpp 01_ArrayInsertion_StaticArray.cpp 02_ArrayInsertion_DynamicArrays.cpp 03_ArrayDeletion_StaticArrays.cpp
DSlabActivity > 03_ArrayDeletion_StaticArrays.cpp > show(int[], int)
1 #include<stdio.h>
2
3
4 void show(int arr[], int n){
5
6     for (int i = 0; i < n; i++)
7     {
8         printf("%d ", arr[i]);
9     }
10    printf("\n");
11 }
12
13 void delt(int *La, int n, int k)
14 {
15     for (int i = k; i < n-1; i++)
16     {
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity" & if ($?) { g++ 02_ArrayInsertion_DynamicArrays.cpp -o 02_ArrayInsertion_DynamicArrays }
6 56 45 5 234
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity" & if ($?) { g++ 03_ArrayDeletion_StaticArrays.cpp -o 03_ArrayDeletion_StaticArrays }
7 8 12 27 88
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity> 
8 12 27 88
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity> 
Ln 7, Col 6 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 204 PM 01-Sep-21

```

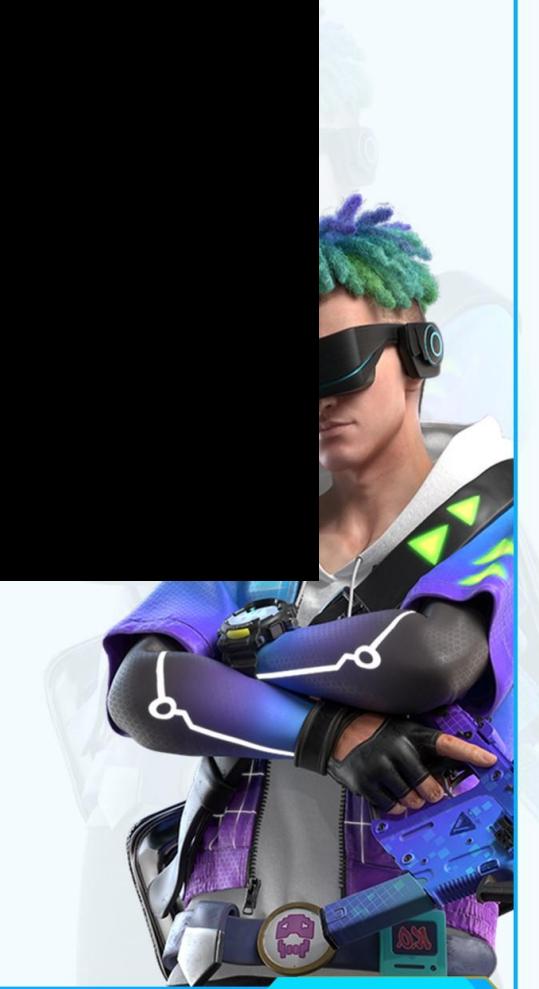
03_ArrayDeletion_DynamicArrays

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

```
void show(int *arr, int n){  
  
    for (int i = 0; i < n; i++)  
    {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
}
```

```
void delt(int *La, int n, int k)  
{  
    for (int i = k; i < n-1; i++)  
    {  
        La[i] = La[i + 1];  
    }  
}
```

```
int main()  
{  
    int * arr = (int*) malloc(100*sizeof(int));  
    arr[0] = 6;  
    arr[1] = 56;  
    arr[2] = 45;  
    arr[3] = 5;  
    arr[4] = 234;  
    arr[5] = 55;  
    int size = 5, index = 5;  
    show(arr, size);  
    delt(arr, size, index);  
    size -= 1;  
    show(arr, size);  
    return 0;  
}
```



```

File Edit Selection View Go Run Terminal Help 04_ArrayDeletion_DynamicArrays.cpp - DS-ALGO - Visual Studio Code
EXPLORER ... ArrayInsertion_StaticArray.cpp 02_ArrayInsertion_DynamicArrays.cpp 03_ArrayDeletion_StaticArrays.cpp 04_ArrayDeletion_DynamicArrays.cpp > ...
OPEN EDITORS ... 01_multiplicationOfMatrix.cpp 02_transposeOfMatrix.cpp 01_fibRec.cpp DSlabActivity 01_ArrayInsertion_StaticArr...
02_factRec.cpp DSlabActivity 03_gcdRec.cpp DSlabActivity 01_ArrayInsertion_Dyna... 03_ArrayDeletion_StaticArr...
04_ArrayDeletion_Dyna... < 04_ArrayDeletion_Dyna...
DS-ALGO ... > 70-80 Binary Search Trees
> 078_dSAlgoLab DS labAssignment1 < DSlabActivity 01_ArrayInsertion_StaticArr...
02_ArrayInsertion_StaticArr... 01.PNG 02_ArrayInsertion_DynamicArr... 02.PNG 03_ArrayDeletion_StaticArr...
03.PNG 04_ArrayDeletion_DynamicArr... 04.PNG > DSlabActivity 15 June > OUTLINE < 04_ArrayDeletion_DynamicArr...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity\" ; if ($?) { g++ 04_ArrayDeletion_DynamicArr...
iArrays.cpp ; if ($?) { ./04_ArrayDeletion_DynamicArr...
6 56 45 5 234
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\DSlabActivity> []
Ln 15, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 RP 01-Sep-21 2:05 PM

```

Lab Assignment 5

01_stack

```

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

int stack[50];
int top=-1;

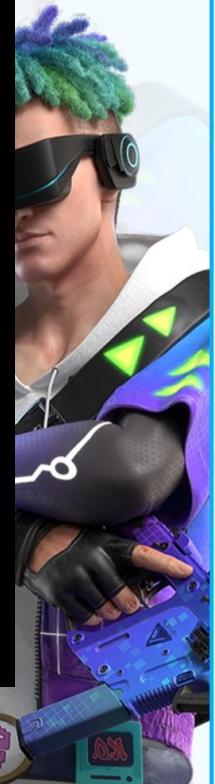
// insertion
void insertion(int num)
{
    if(top==49)
    {
        printf("\nStack is full !!!!\n\n");
        return;
    }
}

```

```
        else
        {
            top=top+1;
            stack[top]=num;
        }
    }
```

```
// deletion
void deletion()
{
    if (top == -1)
    {
        printf("\nStack is already empty !!!!\n");
        return;
    }
    else
    {
        int temp=stack[top];
        top = top - 1;
        printf("\nElement deleted is %d\n", temp);
    }
}
```

```
// display
void display()
{
    if (top == -1)
    {
        printf("\nStack is empty !!!!\n");
        return;
    }
    else
    {
        for(int i=0; i<=top; i++)
        {
            printf("\n%d", stack[i]);
        }
        printf("\n");
    }
}
```

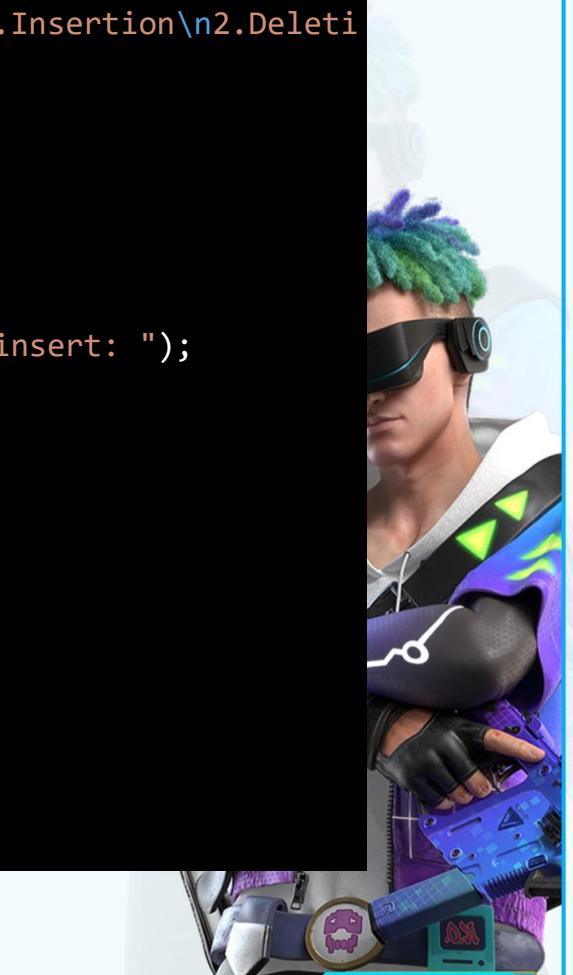


```
// peek
void peek()
{
    if (top == -1)
    {
        printf("\nStack is empty !!!\n");
        return;
    }
    else
    {
        printf("\n%d\n", stack[top]);
    }
}
```

```
int main()
{
    char ch[5]="yes";
    do
    {
        int choice;
        printf("\nSelect your choice:\n1.Insertion\n2.Deletion\n3.Display\n4.Peek\n5.exit\n");
        scanf("%d",&choice);
        switch (choice)
        {
            case 1:
                int num;
                printf("Enter the number to insert: ");
                scanf("%d", &num);
                insertion(num);
                break;

            case 2:
                deletion();
                break;

            case 3:
                display();
                break;
        }
    }
}
```



```

        case 4:
            peek();
            break;

    case 5:
        printf("\nTerminating the program !!!");
        exit(1);
        break;

    default:
        printf("Wrong choice !!!!!!!\n");

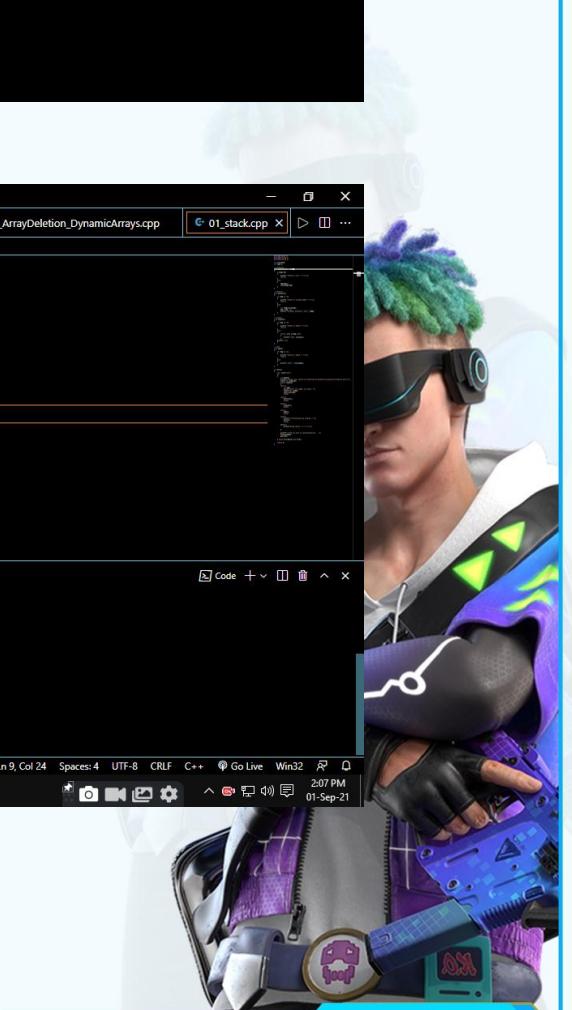
    }

printf("\n\nDo you want to continue(yes/no):   ");
fflush(stdin);
gets(ch);

} while (strcmp(ch,"yes")==0);

return 0;
}

```



A screenshot of Visual Studio Code showing a C++ file named "01_stack.cpp". The code implements a stack using an array. It includes a menu selection loop where users can choose to insert, display, peek, or exit. The terminal shows the execution of the program and user input.

```

File Edit Selection View Go Run Terminal Help
01_stack.cpp - DS-ALGO - Visual Studio Code
ray.cpp 02_ArrayInsertion_DynamicArrays.cpp 03_ArrayDeletion_StaticArrays.cpp 04_ArrayDeletion_DynamicArrays.cpp 01_stack.cpp x
OPEN EDITORS
EXPLORER ... ray.cpp 02_ArrayInsertion_DynamicArrays.cpp 03_ArrayDeletion_StaticArrays.cpp 04_ArrayDeletion_DynamicArrays.cpp 01_stack.cpp x
Semester End Term Project > 01_Stack & Queue using Array > 01_stack.cpp > insertion(int)
1 #include<stdio.h>
2 #include<iostream.h>
3 #include<stdlib.h>
4
5 int stack[50];
6 int top=-1;
7
8 // insertion
9 void insertion(int num)
10 {
11     if(top==49)
12     {
13         printf("\nStack is full !!!\n\n");
14         return;
15     }
16     else
17     {
18         top++;
19         stack[top]=num;
20     }
21 }
22
23 // display
24 void display()
25 {
26     int i;
27     for(i=top;i>=0;i--)
28     {
29         cout<<stack[i]<<endl;
30     }
31 }
32
33 // peek
34 int peek()
35 {
36     if(top<0)
37     {
38         cout<<"Stack is empty";
39         return -1;
40     }
41     else
42     {
43         return stack[top];
44     }
45 }
46
47 // exit
48 void exit()
49 {
50     cout<<"Program Terminated";
51 }
52
53 // main function
54 int main()
55 {
56     int choice;
57     cout<<"Select your choice:
58     1.Insertion
59     2.Display
60     3.Peek
61     4.Exit
62     5.Peek
63     6.Exit
64     ";
65     cin>>choice;
66
67     switch(choice)
68     {
69         case 1:
70             insertion();
71             break;
72         case 2:
73             display();
74             break;
75         case 3:
76             peek();
77             break;
78         case 4:
79             exit();
80             break;
81         case 5:
82             peek();
83             break;
84         case 6:
85             exit();
86             break;
87         default:
88             cout<<"Wrong choice !!!!!!!\n";
89     }
90
91     return 0;
92 }

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

ack.cpp 01_stack.cpp ; if (\$?) { ./01_stack }

Select your choice:

1.Insertion
2.Display
3.Peek
4.Exit
5.Peek
6.Exit

Stack is already empty !!!!

Do you want to continue(yes/no): []

Ln 9, Col 24 Spaces: 4 UTF-8 C++ Go Live Win32 2:07 PM 01-Sep-21

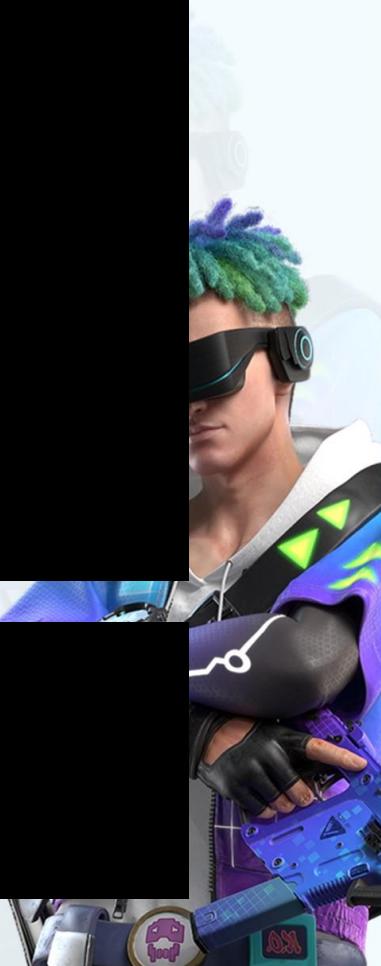
02_queue

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

int queue[50];
int front = -1;
int rear = -1;
```

```
// insertion
void insertion(int num)
{
    if (rear == 49)
    {
        printf("\nQueue is full !!!!\n\n");
        return;
    }
    else if(front== -1 && rear== -1)
    {
        rear++;
        front++;
        queue[rear] = num;
    }
    else
    {
        rear++;
        queue[rear] = num;
    }
}
```

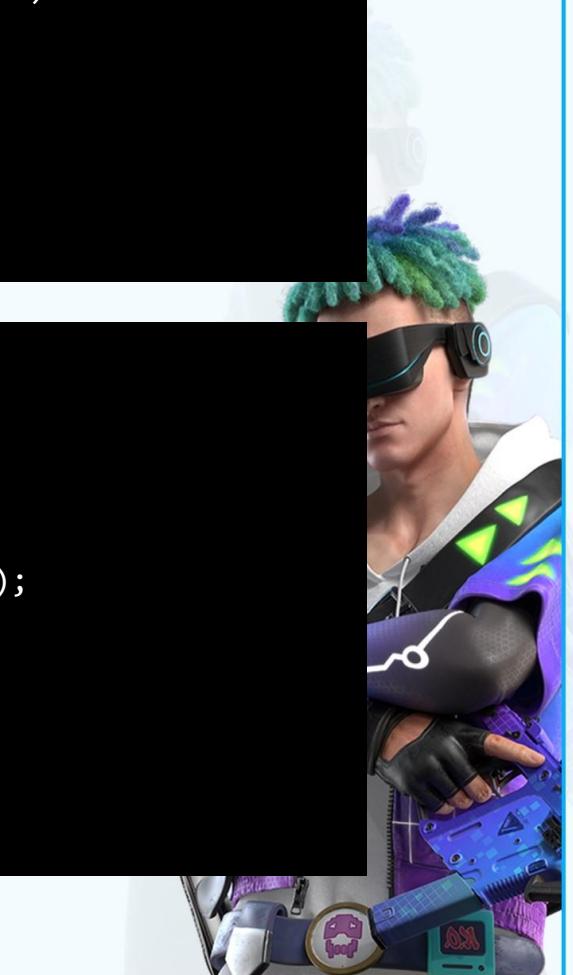
```
// deletion
void deletion()
{
    if (front == -1)
    {
        printf("\nQueue is already empty !!!!\n");
    }
}
```



```
        return;
    }
    else
    {
        int temp = queue[front];
        front++;
        printf("\nElement deleted is %d\n", temp);
    }
}
```

```
// display
void display()
{
    if (front == -1)
    {
        printf("\nQueue is empty !!!!\n");
        return;
    }
    else
    {
        for (int i = front; i <= rear; i++)
        {
            printf("\n%d", queue[i]);
        }
        printf("\n");
    }
}
```

```
// peek
void peek()
{
    if (front == -1)
    {
        printf("\nQueue is empty !!!!\n");
        return;
    }
    else
    {
        printf("\n%d\n", queue[front]);
    }
}
```



```
}
```

```
int main()
{
    char ch[5] = "yes";
    do
    {
        int choice;
        printf("\nSelect your choice:\n1.Insertion\n2.Deletion\n3.Display\n4.Peek\n5.exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                int num;
                printf("Enter the number to insert: ");
                scanf("%d", &num);
                insertion(num);
                break;

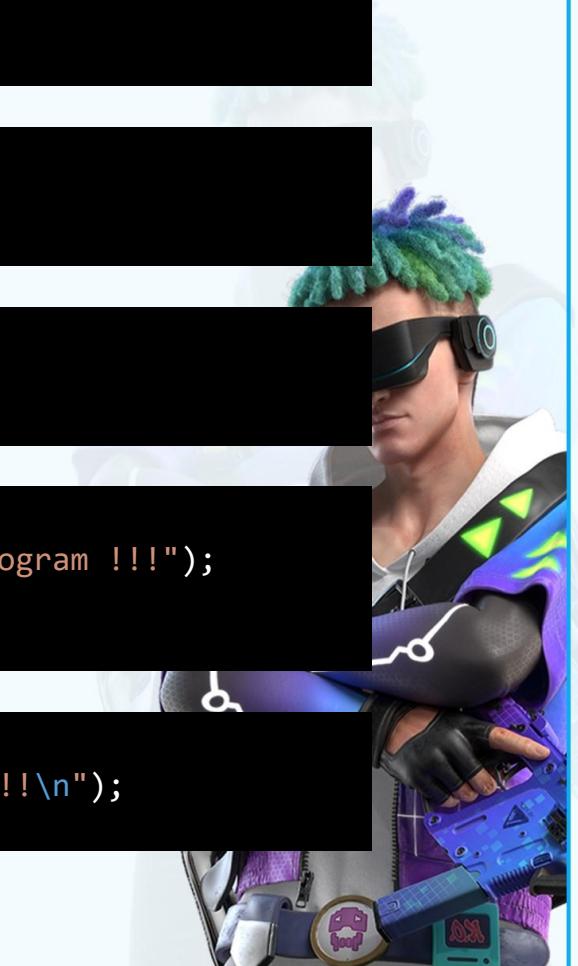
            case 2:
                deletion();
                break;

            case 3:
                display();
                break;

            case 4:
                peek();
                break;

            case 5:
                printf("\nTerminating the program !!!");
                exit(1);
                break;

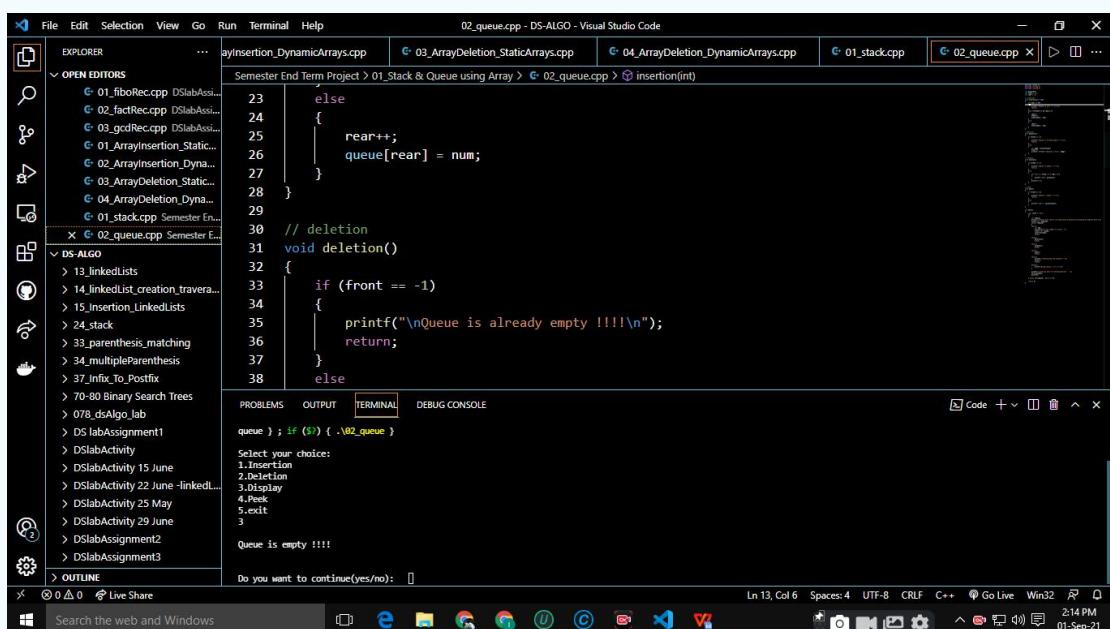
            default:
                printf("Wrong choice !!!!!!!\n");
        }
    }
}
```



```
    printf("\n\nDo you want to continue(yes/no):    ");
    fflush(stdin);
    gets(ch);
```

```
} while (strcmp(ch, "yes") == 0);
```

```
return 0;
}
```



```
File Edit Selection View Go Run Terminal Help 02_queue.cpp - DS-ALGO - Visual Studio Code
OPEN EDITORS ayInsertion_DynamicArrays.cpp 03_ArrayDeletion_StaticArrays.cpp 04_ArrayDeletion_DynamicArrays.cpp 01_stack.cpp 02_queue.cpp ...
Semester End Term Project > 01_Stack & Queue using Array > 02_queue.cpp > insertion(int)
23     else
24     {
25         rear++;
26         queue[rear] = num;
27     }
28 }
29
30 // deletion
31 void deletion()
32 {
33     if (front == -1)
34     {
35         printf("\nQueue is already empty !!!!\n");
36         return;
37     }
38 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
queue ) ; if ( $ ) { .\02_queue }

Select your choice:
1.Insertion
2.Deletion
3.Display
4.Peek
5.Exit
3

Queue is empty !!!!
```

Do you want to continue(yes/no):

Ln 13, Col 6 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 2:14 PM 01-Sep-21

03_circularQueue

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define max 50

int queue[max];
int front = -1;
```

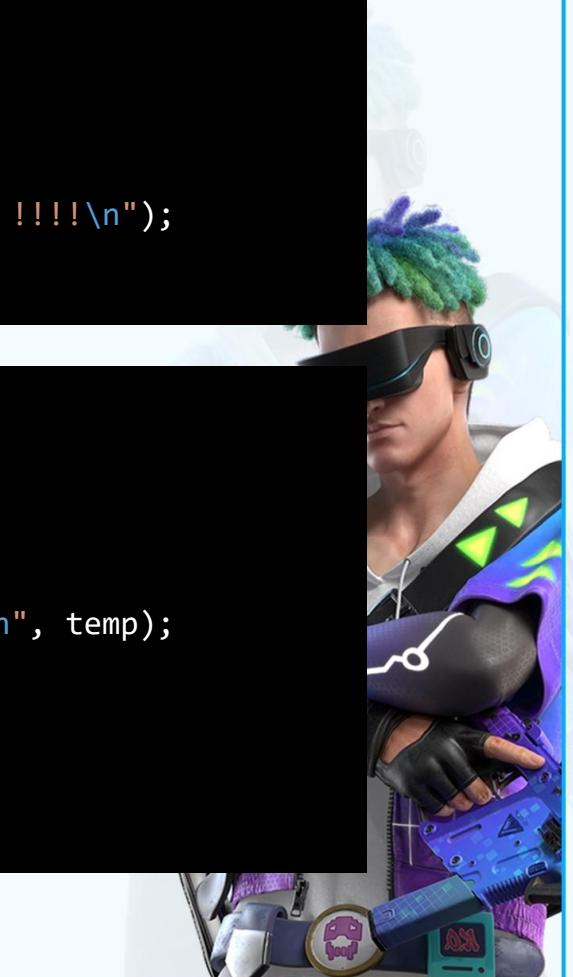
```
int rear = -1;

// insertion
void insertion(int num)
{
    if ((rear+1)%(max-1) == front)
    {
        printf("\nQueue is full !!!!\n\n");
        return;
    }
    else if (front == -1)
    {
        front++;
    }

    rear = (rear+1)%(max-1);
    queue[rear] = num;
}
```

```
// deletion
void deletion()
{
    if (front == -1)
    {
        printf("\nQueue is already empty !!!!\n");
        return;
    }
```

```
else if(front==rear)
{
    int temp = queue[front];
    front=-1;
    rear=-1;
    printf("\nElement deleted is %d\n", temp);
}
else
{
    int temp = queue[front];
    front++;
```



```
        printf("\nElement deleted is %d\n", temp);
    }
}
```

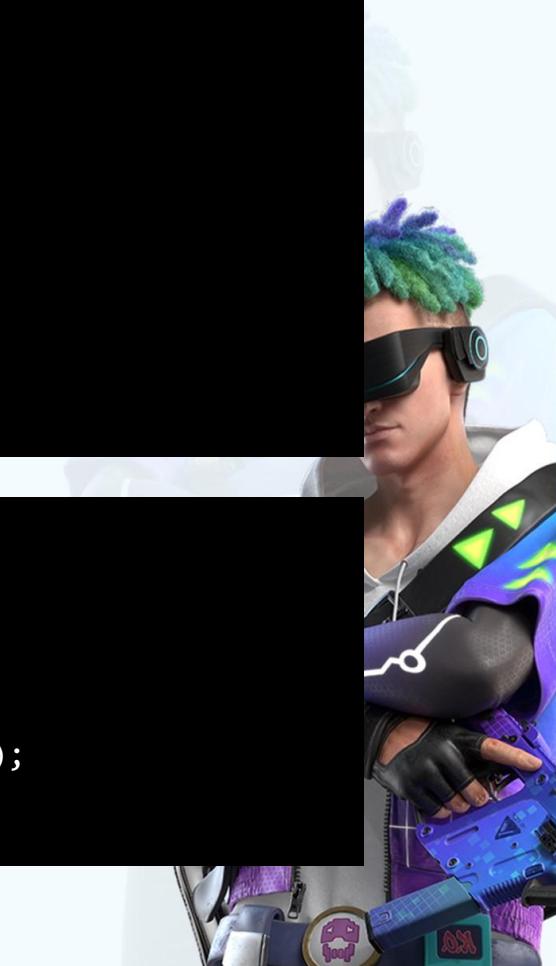
```
// display
void display()
{
    printf("\nFront = %d\nRear = %d\n", front, rear);
```

```
    if (front == -1)
    {
        printf("\nQueue is empty !!!!\n");
        return;
    }
```

```
    else if(front==rear)
    {
        printf("\n%d\n", queue[front]);
    }
```

```
else
{
    int i = front;
    while(i!=rear)
    {
        printf("\n%d", queue[i]);
        i=(i+1)% (max-1);
    }
    printf("\n%d\n", queue[i]);
}
```

```
// peek
void peek()
{
    if (front == -1)
    {
        printf("\nQueue is empty !!!!\n");
        return;
    }
```



```
        else
        {
            printf("\n%d\n", queue[front]);
        }
    }
```

```
int main()
{
    char ch[5] = "yes";
    do
    {
        int choice;
        printf("\nSelect your choice:\n1.Insertion\n2.Deletion\n3.Display\n4.Peek\n5.exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                int num;
                printf("Enter the number to insert: ");
                scanf("%d", &num);
                insertion(num);
                break;

            case 2:
                deletion();
                break;

            case 3:
                display();
                break;

            case 4:
                peek();
                break;

            case 5:
                printf("\nTerminating the program !!!");
                exit(1);
                break;
        }
    }
}
```

```
default:  
    printf("Wrong choice !!!!!!!\n");  
}
```

```
printf("\n\nDo you want to continue(yes/no): ");  
fflush(stdin);  
gets(ch);
```

```
} while (strcmp(ch, "yes") == 0);
```

```
return 0;  
}
```



A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "03_circularQueue.cpp - DS-ALGO - Visual Studio Code". The Explorer sidebar on the left lists various C++ files and projects. The main editor area displays the code for "03_circularQueue.cpp". The code implements a circular queue with insertion and checking if it's full. The terminal tab at the bottom shows the execution of the program and its output. The status bar at the bottom right shows the current file is "03_circularQueue.cpp", has 14 lines, 6 columns, and is in UTF-8 encoding.

```
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#define max 50  
  
int queue[max];  
int front = -1;  
int rear = -1;  
  
// insertion  
void insertion(int num)  
{  
    if ((rear+1)%max-1 == front)  
    {  
        printf("\nQueue is full !!!\n\n");  
        return;  
    }  
    else  
    {  
        queue[rear+1] = num;  
        rear = (rear+1)%max-1;  
        printf("Element inserted successfully\n");  
    }  
}  
  
int deletion()  
{  
    if (front == -1)  
    {  
        printf("Queue is empty !!!\n");  
        return;  
    }  
    else  
    {  
        int num = queue[front];  
        queue[front] = -1;  
        front = (front+1)%max-1;  
        return num;  
    }  
}  
  
int peek()  
{  
    if (front == -1)  
    {  
        printf("Queue is empty !!!\n");  
        return;  
    }  
    else  
    {  
        printf("Front element is %d\n", queue[front]);  
        return queue[front];  
    }  
}  
  
int display()  
{  
    if (front == -1)  
    {  
        printf("Queue is empty !!!\n");  
        return;  
    }  
    else  
    {  
        printf("Elements in Queue are :\n");  
        for (int i=front; i<=rear; i=(i+1)%max-1)  
            printf("%d ", queue[i]);  
        printf("\n");  
    }  
}  
  
int main()  
{  
    int choice;  
    do  
    {  
        printf("Select your choice:  
1.Insertion  
2.Deletion  
3.Display  
4.Peek  
5.exit  
4  
        Queue is empty !!!  
        Do you want to continue(yes/no): ");  
        scanf("%c", &choice);  
        switch (choice)  
        {  
            case '1':  
                insertion();  
                break;  
            case '2':  
                deletion();  
                break;  
            case '3':  
                display();  
                break;  
            case '4':  
                peek();  
                break;  
            case '5':  
                exit(0);  
                break;  
            default:  
                printf("Wrong choice !!!!!!!\n");  
        }  
    } while (strcmp(choice, "yes") == 0);  
    return 0;  
}
```

Lab Assignment 6

01_queueOperations

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

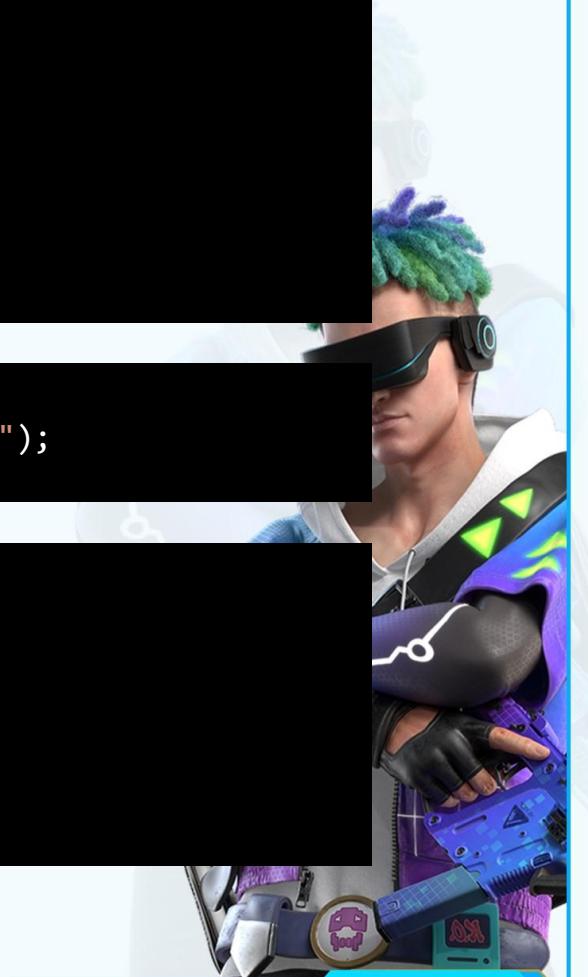
int queue[50];
int front = -1;
int rear = -1;
```

```
// insertion
void insertion(int item)
{
    if(rear<49)
    {
        if(front== -1)
            front++;

        rear++;
        queue[rear]=item;
    }

    else
        printf("\nThe queue is full !!!!");
}
```

```
// deletion
void deletion()
{
    int item;
    if (front != -1)
    {
        item=queue[front];
```



```

        if(front==rear)
        {
            front=-1;
            rear=-1;
        }

    else
        front++;



        printf("\nNumber deleted is %d\n", item);
    }
else
{
    printf("\nQueue is already empty !!!!\n");
}
}

```

```

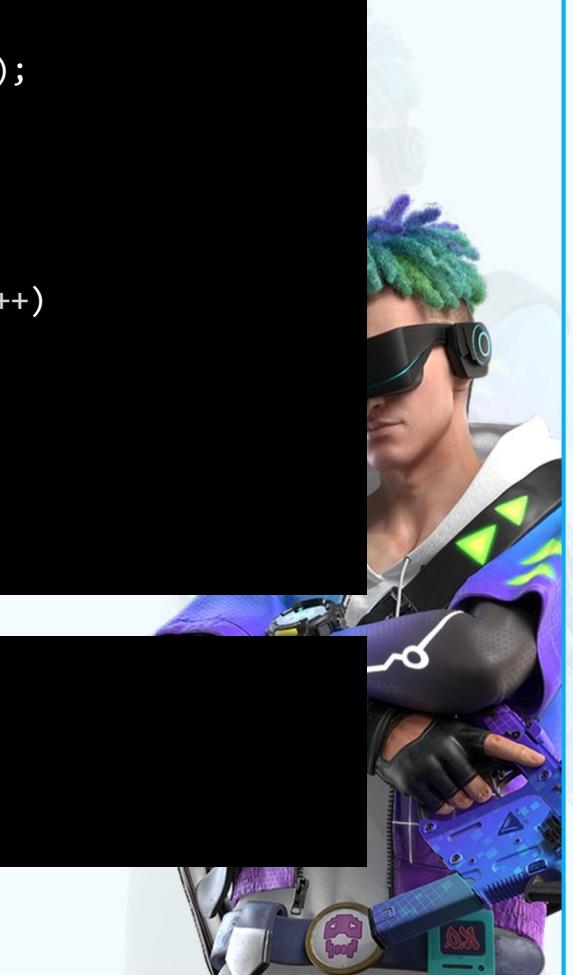
// display
void display()
{
    if (front == -1)
    {
        printf("\nQueue is empty !!!!\n");
        return;
    }
    else
    {
        for (int i = front; i <= rear; i++)
        {
            printf("\n%d", queue[i]);
        }
        printf("\n");
    }
}

```

```

// peek
void peek()
{
    if (front == -1)
    {

```



```
        printf("\nQueue is empty !!!!\n");
        return;
    }
    else
    {
        printf("\n%d\n", queue[front]);
    }
}
```

```
int main()
{
    char ch[5] = "yes";
    do
    {
        int choice;
        printf("\nSelect your choice:\n1.Insertion\n2.Deletion\n3.Display\n4.Peek\n5.exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                int num;
                printf("Enter the number to insert: ");
                scanf("%d", &num);
                insertion(num);
                break;

            case 2:
                deletion();
                break;

            case 3:
                display();
                break;

            case 4:
                peek();
                break;

            case 5:
```

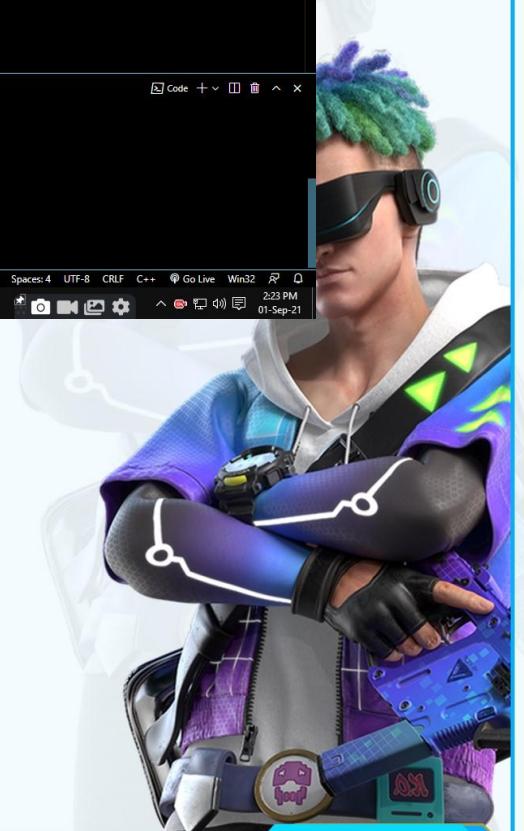
```
        printf("\nTerminating the program !!!");
        exit(1);
        break;
```

```
    default:
        printf("Wrong choice !!!!!!!\n");
    }
```

```
printf("\n\nDo you want to continue(yes/no):   ");
fflush(stdin);
gets(ch);
```

```
} while (strcmp(ch, "yes") == 0);
```

```
return 0;
}
```



A screenshot of Visual Studio Code showing the code for a queue operations program. The code includes functions for insertion, deletion, and display, along with a loop for user interaction. The interface shows multiple tabs open, including files for arrays, stacks, and queues.

```
File Edit Selection View Go Run Terminal Help 01_queueOperations.cpp - DS ALGO - Visual Studio Code
arrays.cpp 04_ArrayDeletion_DynamicArrays.cpp 01_stack.cpp 02_queue.cpp 03_circularQueue.cpp 01_queueOperations.cpp ...
DSLabActivity 25 May > 01.queueOperations.cpp > ...
OPEN EDITORS
03_godRec.cpp DSlabAss...
01_ArrayInsertion_Static...
02_ArrayInsertion_Dyna...
03_ArrayDeletion_Static...
04_ArrayDeletion_Dyna...
01_stack.cpp Semester End...
02_queue.cpp Semester End...
03_circularQueue.cpp Semester End...
x 01_queueOperations.cpp ...
DS ALGO
DSlabActivity 15 June
> DSlabActivity 22 June -linked...
DSlabActivity 25 May
01_queueOperations.cpp
01_queueOperations.exe
> DSlabActivity 29 June
> DSlabAssignment2
> DSlabAssignment3
> lab manual 8 jun
> linked list polynomial
> other programs
Semester End Term Project 0...
01_stack.cpp
01_stack.exe
02_queue.cpp
02_queue.exe
03_circularQueue.cpp
=03_circularQueue.exe ...
OUTLINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
5.exit
1
Enter the number to insert: 4

Do you want to continue(yes/no): yes
Select your choice:
1.Insert
2.Deletion
3.Display
4.Peek
5.Exit
Ln 9, Col 13  Spaces:4  UTF-8  CRLF  C++  Go Live  Win32  2:23 PM  01-Sep-21
```

Lab Assignment 7

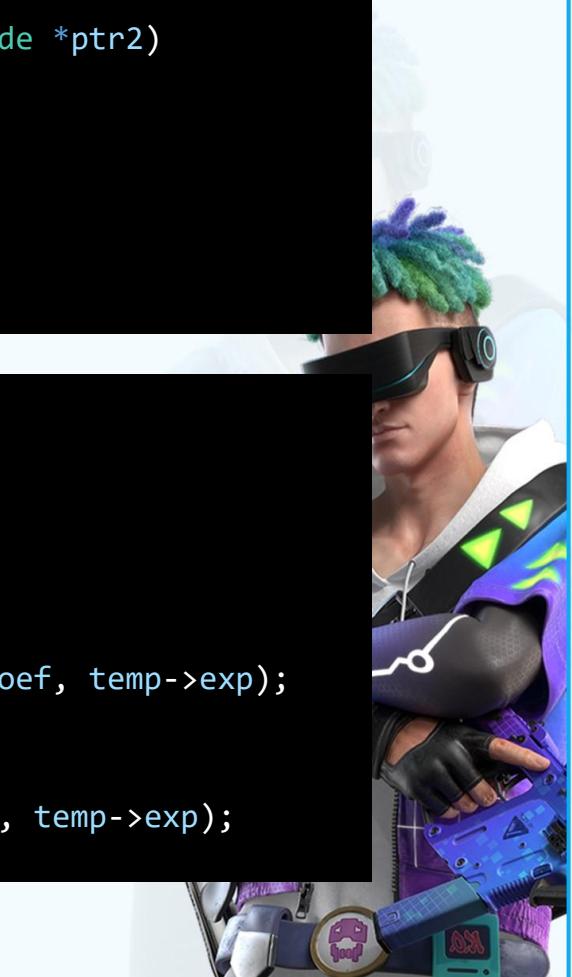
01_linkedList_polynomial

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

struct node
{
    int exp;
    float coef;
    struct node *next;
};

// function append
// appends ptr1 to ptr2
void append(struct node *ptr1, struct node *ptr2)
{
    struct node *ptr = ptr2;
    while (ptr->next != NULL)
        ptr = ptr->next;
    ptr->next = ptr1;
}

// function for traversal
void traversal(struct node *head)
{
    struct node *temp = head;
    while (temp->next != NULL)
    {
        printf("%4.1f(x)^%d + ", temp->coef, temp->exp);
        temp = temp->next;
    }
    printf("%4.1f(x)^%d\n\n", temp->coef, temp->exp);
}
```



```

// function for insertion
struct node *insertion(struct node *head, int ex, float cof)
{
    struct node *temp = (struct node *)malloc(sizeof(struct
node));
    // printf("%4.2fx^%d\n", cof, ex);
    temp->coef = cof;
    temp->exp = ex;
    temp->next = NULL;
    // printf("%4.2fx^%d\n", temp->coef, temp->exp);
    if (head == NULL)
        head = temp;
    else
    {
        struct node *ptr = head;
        while (ptr->next != NULL)
            ptr = ptr->next;

        ptr->next = temp;
    }
    return head;
}

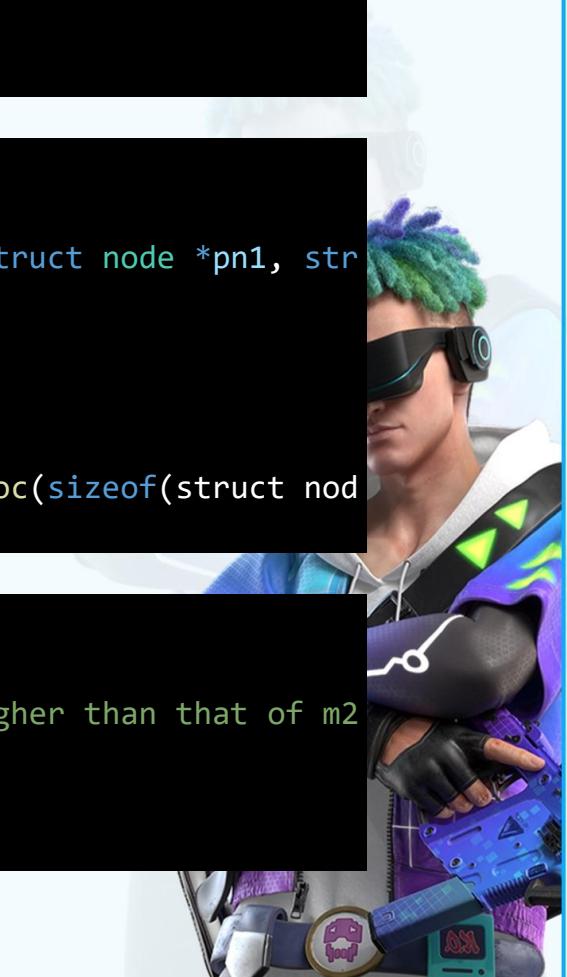
```

```

// function to add two polynomials
// pn1 + pn2 = pn3
struct node *addPoly(struct node *pn3, struct node *pn1, str
uct node *pn2)
{
    struct node *m1 = pn1;
    struct node *m2 = pn2;
    struct node *m3 = (struct node*)malloc(sizeof(struct nod
e));

    while((m1!=NULL) || (m2!=NULL))
    {
        // when power of m1's term is higher than that of m2
's
        while(m1->exp > m2->exp)
        {

```



```
        m3->exp = m1->exp;
        m3->coef = m1->coef;
        append(m3,pn3);
        m1 = m1->next;
    }

    // when power of m2's term is higher than that of m1
's
    while (m1->exp < m2->exp)
    {
        m3->exp = m2->exp;
        m3->coef = m2->coef;
        append(m3,pn3);
        m2 = m2->next;
    }

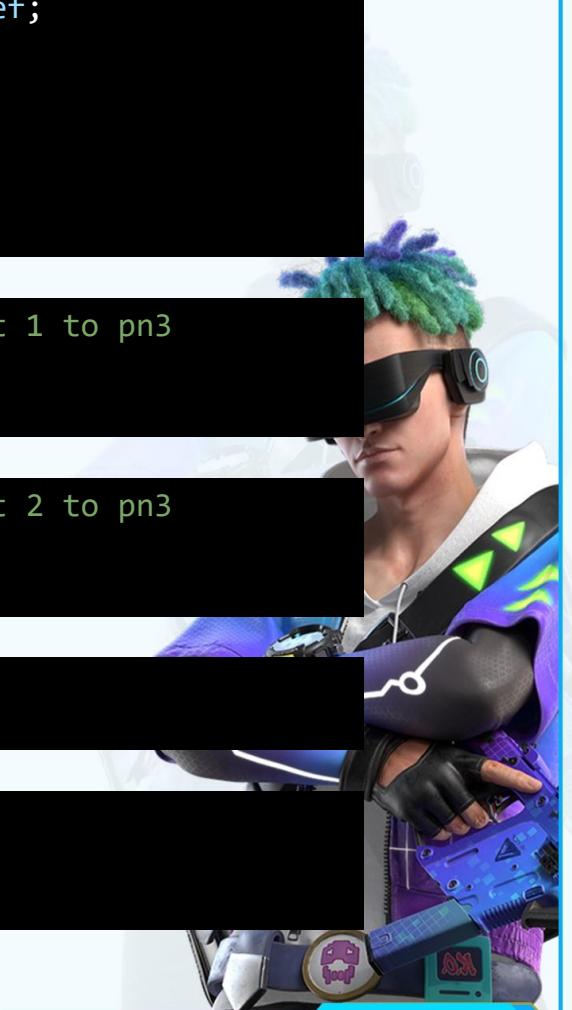
    // when power of m2's term is equal to that of m1's
    while (m1->exp == m2->exp)
    {
        m3->exp = m2->exp;
        m3->coef = m2->coef + m1->coef;
        append(m3,pn3);
        m2 = m2->next;
        m1 = m1->next;
    }
}

// if list 2 exhausts, appending list 1 to pn3
if(m1!=NULL)
    append(m1, pn3);

// if list 1 exhausts, appending list 2 to pn3
else
    append(m2, pn3);

return pn3;
}

int main()
{
    char ch = 'y';
}
```



```
struct node *pn1 = NULL, *pn2 = NULL, *pn3 = NULL;
```

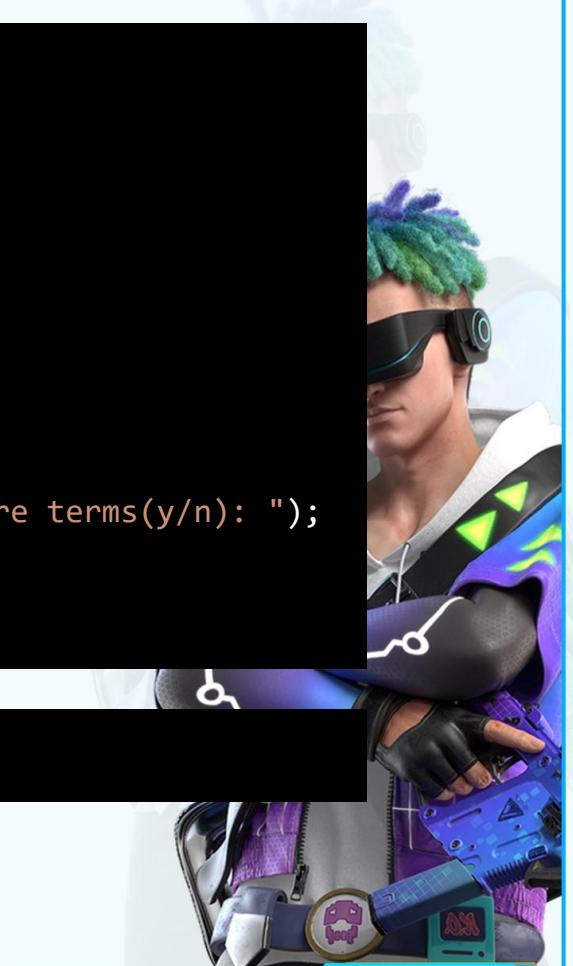
```
printf("Polynomial 1: \n");
while (ch == 'y')
{
    int ex;
    float cof;
    printf("Enter Coefficient : ");
    scanf("%f", &cof);
    printf("Enter Exponent : ");
    scanf("%d", &ex);
    pn1 = insertion(pn1, ex, cof);
    printf("Do you want to insert more terms(y/n): ");
    fflush(stdin);
    scanf("%c", &ch);
}
```

```
printf("The polynomial 1 is : ");
traversal(pn1);
```

```
ch = 'y'; // reinitialising y = 'y'
```

```
printf("Polynomial 2: \n");
while (ch == 'y')
{
    int ex;
    float cof;
    printf("Enter Coefficient : ");
    scanf("%f", &cof);
    printf("Enter Exponent : ");
    scanf("%d", &ex);
    pn2 = insertion(pn2, ex, cof);
    printf("Do you want to insert more terms(y/n): ");
    fflush(stdin);
    scanf("%c", &ch);
}
```

```
printf("The polynomial 2 is : ");
traversal(pn2);
```



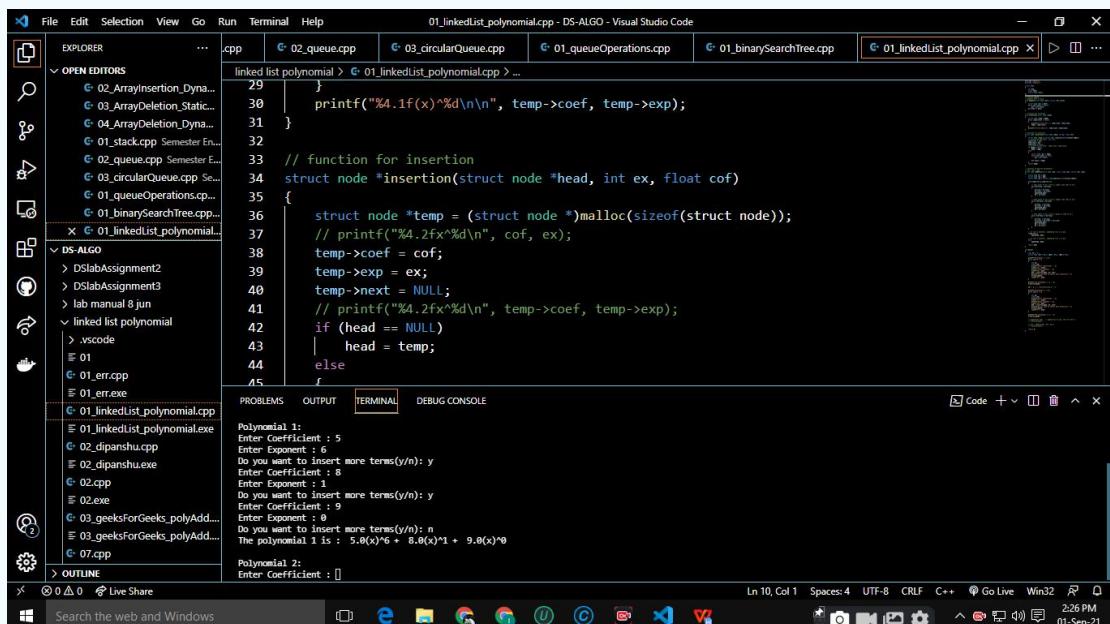
```

        // append(pn1,pn2); // append pn1 to pn2, just for test
    :)
    // traversal(pn2);

    // pn3 = addPoly(pn3, pn2, pn1);
    // traversal(pn3);

    return 0;
}

```



```

File Edit Selection View Go Run Terminal Help 01_LinkedList_polynomial.cpp - DS-ALGO - Visual Studio Code
EXPLORER OPEN EDITORS .cpp 02_queue.cpp 03_circularQueue.cpp 01_queueOperations.cpp 01_binarySearchTree.cpp 01_LinkedList_polynomial.cpp ...
linked list polynomial > 01_LinkedList_polynomial.cpp ...
29 }
30     printf("%4.1f(x)^%d\n\n", temp->coef, temp->exp);
31 }
32
33 // function for insertion
34 struct node *insertion(struct node *head, int ex, float cof)
35 {
36     struct node *temp = (struct node *)malloc(sizeof(struct node));
37     // printf("%4.2fx^%d\n", cof, ex);
38     temp->coef = cof;
39     temp->exp = ex;
40     temp->next = NULL;
41     // printf("%4.2fx^%d\n", temp->coef, temp->exp);
42     if (head == NULL)
43     {
44         head = temp;
45     }
46     else
47     {
48         struct node *curr = head;
49         while (curr->next != NULL)
50         {
51             curr = curr->next;
52         }
53         curr->next = temp;
54     }
55 }
56
57 void printPoly(struct node *head)
58 {
59     if (head == NULL)
60     {
61         printf("The polynomial is empty.");
62     }
63     else
64     {
65         struct node *temp = head;
66         while (temp != NULL)
67         {
68             printf("%4.1f(x)^%d ", temp->coef, temp->exp);
69             temp = temp->next;
70         }
71     }
72 }
73
74 int main()
75 {
76     struct node *head1 = NULL;
77     struct node *head2 = NULL;
78
79     insertion(head1, 6, 5.0);
80     insertion(head1, 1, 8.0);
81     insertion(head1, 0, 9.8);
82
83     insertion(head2, 6, 3.0);
84     insertion(head2, 1, 2.0);
85
86     struct node *pn1 = addPoly(head1, head2);
87
88     printPoly(pn1);
89
90     free(head1);
91     free(head2);
92
93     return 0;
94 }

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Polyomial 1:
Enter Coefficient : 5
Enter Exponent : 6
Do you want to insert more terms(y/n): y
Enter Coefficient : 8
Enter Exponent : 1
Do you want to insert more terms(y/n): y
Enter Coefficient : 9
Enter Exponent : 0
Do you want to insert more terms(y/n): n
The polynomial 1 is : 5.0(x)^6 + 8.0(x)^1 + 9.8(x)^0

Polyomial 2:
Enter Coefficient : 0

Ln 10, Col 1 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 2:26 PM 01-Sep-21

03_geeksForGeeks_polyAdd

```

#include <bits/stdc++.h>
using namespace std;

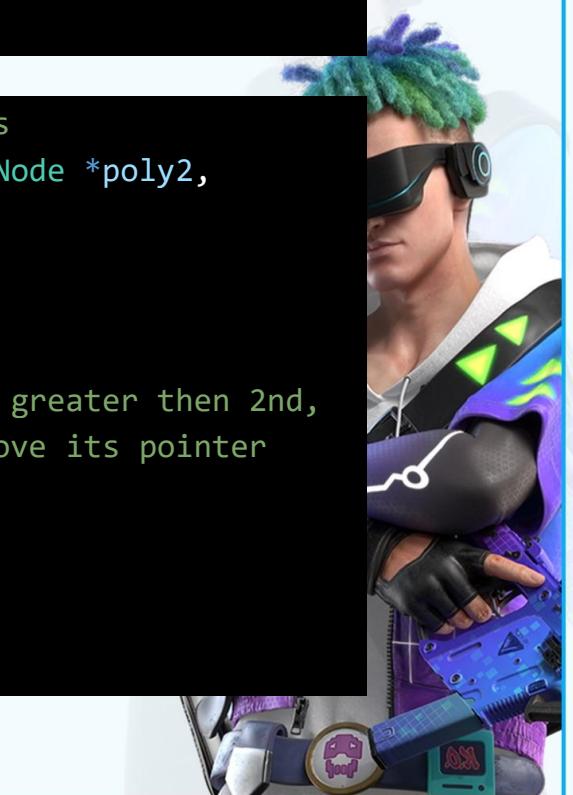
// Node structure containing power and coefficient of
// variable
struct Node
{
    int coeff;
    int pow;
    struct Node *next;
}

```

```
};
```

```
// Function to create new node
void create_node(int x, int y, struct Node **temp)
{
    struct Node *r, *z;
    z = *temp;
    if (z == NULL)
    {
        r = (struct Node *)malloc(sizeof(struct Node));
        r->coeff = x;
        r->pow = y;
        *temp = r;
        r->next = (struct Node *)malloc(sizeof(struct Node));
        r = r->next;
        r->next = NULL;
    }
    else
    {
        r->coeff = x;
        r->pow = y;
        r->next = (struct Node *)malloc(sizeof(struct Node));
        r = r->next;
        r->next = NULL;
    }
}
```

```
// Function Adding two polynomial numbers
void polyadd(struct Node *poly1, struct Node *poly2,
             struct Node *poly)
{
    while (poly1->next && poly2->next)
    {
        // If power of 1st polynomial is greater than 2nd,
        // then store 1st as it is and move its pointer
        if (poly1->pow > poly2->pow)
        {
            poly->pow = poly1->pow;
            poly->coeff = poly1->coeff;
            poly1 = poly1->next;
        }
        else
        {
            poly->pow = poly2->pow;
            poly->coeff = poly2->coeff;
            poly2 = poly2->next;
        }
        poly->next = (struct Node *)malloc(sizeof(struct Node));
        poly = poly->next;
    }
}
```



```
}
```

```
// If power of 2nd polynomial is greater than 1st,  
// then store 2nd as it is and move its pointer  
else if (poly1->pow < poly2->pow)  
{  
    poly->pow = poly2->pow;  
    poly->coeff = poly2->coeff;  
    poly2 = poly2->next;  
}
```

```
// If power of both polynomial numbers is same then  
// add their coefficients  
else  
{  
    poly->pow = poly1->pow;  
    poly->coeff = poly1->coeff + poly2->coeff;  
    poly1 = poly1->next;  
    poly2 = poly2->next;  
}
```

```
// Dynamically create new node  
poly->next = (struct Node *)malloc(sizeof(struct Node));  
poly = poly->next;  
poly->next = NULL;  
}  
while (poly1->next || poly2->next)  
{  
    if (poly1->next)  
    {  
        poly->pow = poly1->pow;  
        poly->coeff = poly1->coeff;  
        poly1 = poly1->next;  
    }  
    if (poly2->next)  
    {  
        poly->pow = poly2->pow;  
        poly->coeff = poly2->coeff;  
        poly2 = poly2->next;  
    }
```



```

        }
        poly-
>next = (struct Node *)malloc(sizeof(struct Node));
        poly = poly->next;
        poly->next = NULL;
    }
}

```

```

// Display Linked list
void show(struct Node *node)
{
    while (node->next != NULL)
    {
        printf("%dx^%d", node->coeff, node->pow);
        node = node->next;
        if (node->coeff >= 0)
        {
            if (node->next != NULL)
                printf("+");
        }
    }
    printf("\n");
}

```

```

// Driver code
int main()
{
    struct Node *poly1 = NULL, *poly2 = NULL, *poly = NULL;

    // Create first list of 5x^2 + 4x^1 + 2x^0
    create_node(5, 2, &poly1);
    create_node(4, 1, &poly1);
    create_node(2, 0, &poly1);

    // Create second list of -5x^1 - 5x^0
    create_node(-5, 1, &poly2);
    create_node(-5, 0, &poly2);

    printf("1st Number: ");
    show(poly1);
}

```

```
    printf("2nd Number: ");
    show(poly2);
```

```
poly = (struct Node *)malloc(sizeof(struct Node));
```

```
// Function add two polynomial numbers
polyadd(poly1, poly2, poly);
```

```
// Display resultant List
printf("Added polynomial: ");
show(poly);
```

```
return 0;
}
```



A screenshot of Microsoft Visual Studio Code running on a Windows operating system. The code editor displays a C++ program for adding two polynomials. The file being edited is "03_geeksForGeeks_polyAdd.cpp". The terminal window shows the output of the program, which adds two polynomials and prints the result. The status bar at the bottom indicates the file is 0% complete.

```
File Edit Selection View Go Run Terminal Help 03_geeksForGeeks_polyAdd.cpp - DS-ALGO - Visual Studio Code
EXPLORER OPEN EDITORS circularQueue.cpp 01_queueOperations.cpp 01_binarySearchTree.cpp 01_linkedList_polynomial.cpp 03_geeksForGeeks_polyAdd.cpp
linked list polynomial > 03_geeksForGeeks_polyAdd.cpp ...
1 // C++ program for addition of two polynomials
2 // using Linked Lists
3 #include <bits/stdc++.h>
4 using namespace std;
5
6 // Node structure containing power and coefficient of
7 // variable
8 struct Node
9 {
10     int coeff;
11     int pow;
12     struct Node *next;
13 };
14
15 // Function to create new node
16 void create_node(int x, int y, struct Node **temp)
PROBLEMS OUTPUT DEBUG CONSOLE
Windows PowerShell Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\linked list polynomial\" ; if ($_) { g++ 03_geeksForGeeks_polyAdd.cpp --o 03_geeksForGeeks_polyAdd... }
1yAdd : 1. If ( $_) { g++ 03_geeksForGeeks_polyAdd... }
1st 1yAdd: 2+4*x^2+2*x^4
2nd Number: -5*x^1+5*x^3
Added polynomial: 5*x^2-1*x^1-3*x^4
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\linked list polynomial> [REDACTED]
Ln 6, Col 54 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 2:39 PM 01-Sep-21
```

Lab Assignment 8

01_sorting

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

void display(int a[], int n);
void bubble_sort(int a[], int n);
void selection_sort(int a[], int n);
void insertion_sort(int a[], int n);
```

```
int main()
{
    int n, choice, i;
    char ch[20];
    printf("Enter no. of elements u want to sort : ");
    scanf("%d", &n);
    int arr[n];
    for (i = 0; i < n; i++)
    {
        printf("Enter %d Element : ", i + 1);
        scanf("%d", &arr[i]);
    }
    printf("Please select any option Given Below for Sorting
          : \n");

    while (1)
    {

        printf("\n1. Bubble Sort\n2. Selection Sort\n3. Inse
rtion Sort\n4. Display Array.\n5. Exit the Program.\n");
        printf("\nEnter your Choice : ");
        scanf("%d", &choice);
```

```

switch (choice)
{
case 1:
    bubble_sort(arr, n);
    break;
case 2:
    selection_sort(arr, n);
    break;
case 3:
    insertion_sort(arr, n);
    break;
case 4:

    display(arr, n);
    break;

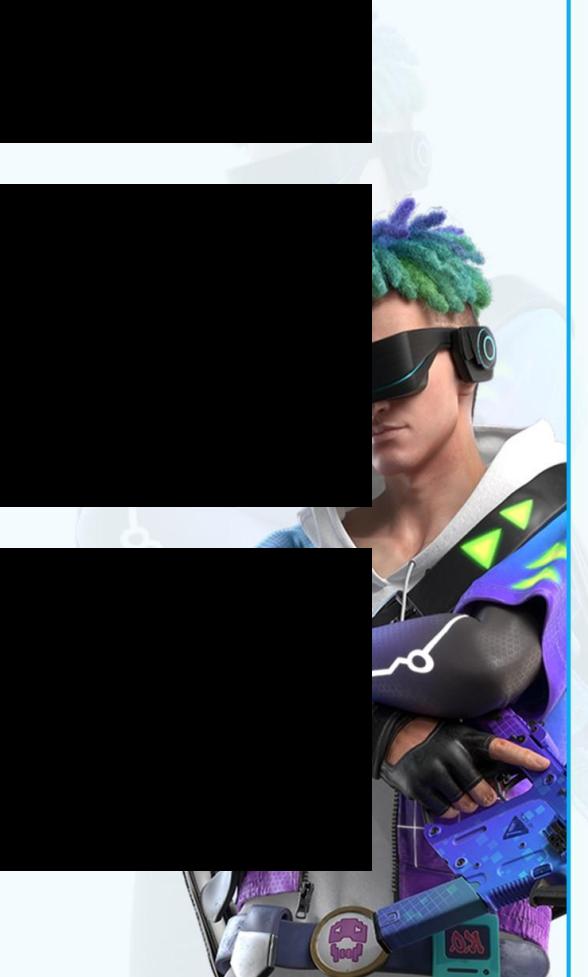
case 5:
    return 0;
default:
    printf("\nPlease Select only 1-5 option ----\n");
}
}

return 0;
}

void display(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf(" %d ", arr[i]);
    }
}

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

```



```

        if (arr[j] > arr[j + 1])
        {
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
printf("After Bubble sort Elements are : ");
display(arr, n);
}

```

```

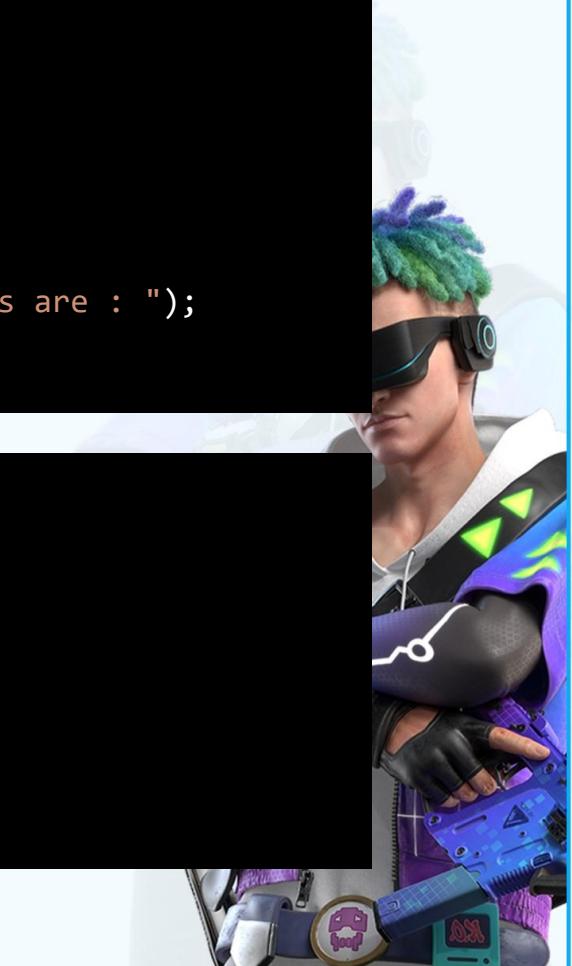
void selection_sort(int arr[], int n)
{
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (arr[i] > arr[j])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    printf("After Selection sort Elements are : ");
    display(arr, n);
}

```

```

void insertion_sort(int arr[], int n)
{
    int i, j, min;
    for (i = 1; i < n; i++)
    {
        min = arr[i];
        j = i - 1;
        while (min < arr[j] && j >= 0)
        {

```



```

        arr[j + 1] = arr[j];
        j = j - 1;
    }
    arr[j + 1] = min;
}
printf("After Insertion sort Elements are : ");
display(arr, n);
}

```



A screenshot of Visual Studio Code showing a C program for insertion sort. The terminal window displays the execution of the program, which sorts an array of four elements (6, 7, 8, 89) using the insertion sort algorithm. The output shows the sorted array as 6, 7, 8, 89.

```

File Edit Selection View Go Run Terminal Help 01_sorting.cpp - programs of C - Visual Studio Code
EXPLORER PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\Lucky\my cpp\programs of C> cd "E:\Lucky\my cpp\programs of C\Divya"
Enter no. of elements u want to sort : 4
Enter 1 Element : 6
Enter 2 Element : 7
Enter 3 Element : 8
Enter 4 Element : 89
Please select any option Given Below for Sorting :
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 3
After Insertion sort Elements are : 6 7 8 89
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 3
After Insertion sort Elements are : 6 7 8 89
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Display Array.
5. Exit the Program.

Enter your Choice : 5
PS E:\Lucky\my cpp\programs of C\Divya> []

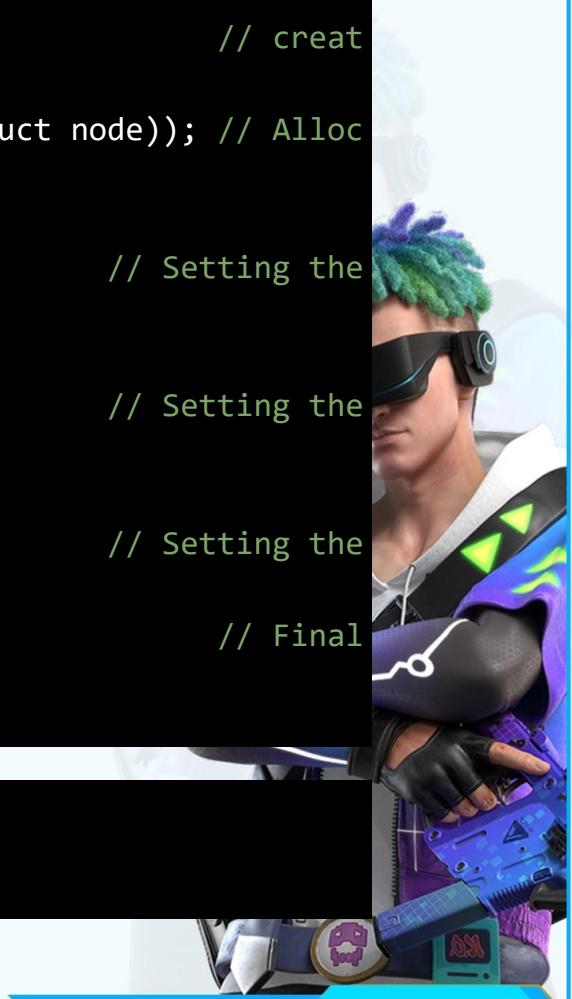
```

Ln 65, Col 20 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 2:35 PM 01-Sep-21

Lab Assignment 9

01_binarySearchTree

```
// checking whether a tree is a BST?  
#include <stdio.h>  
#include <malloc.h>  
  
struct node  
{  
    int data;  
    struct node *left;  
    struct node *right;  
};  
  
struct node *createNode(int data)  
{  
    struct node *n;                                // creating a node pointer  
    n = (struct node *)malloc(sizeof(struct node)); // Allocating memory in the heap  
    n->data = data;                               // Setting the data  
    n->left = NULL;                             // Setting the left and right children to NULL  
    n->right = NULL;                            // Setting the left and right children to NULL  
    return n;                                    // Final  
    // by returning the created node  
}  
  
void preOrder(struct node *root)  
{  
    if (root != NULL)  
    {  
        //  
    }  
}
```

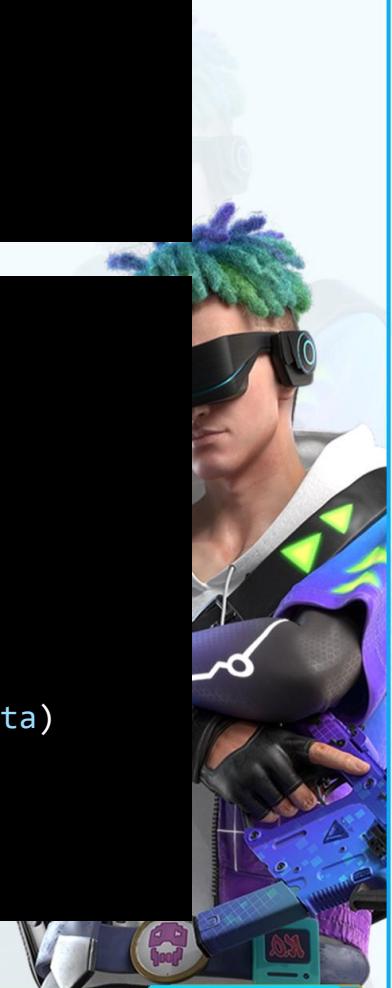


```
    {
        printf("%d ", root->data);
        preOrder(root->left);
        preOrder(root->right);
    }
}
```

```
void postOrder(struct node *root)
{
    if (root != NULL)
    {
        postOrder(root->left);
        postOrder(root->right);
        printf("%d ", root->data);
    }
}
```

```
void inOrder(struct node *root)
{
    if (root != NULL)
    {
        inOrder(root->left);
        printf("%d ", root->data);
        inOrder(root->right);
    }
}
```

```
int isBST(struct node *root)
{
    static struct node *prev = NULL;
    if (root != NULL)
    {
        if (!isBST(root->left))
        {
            return 0;
        }
        if (prev != NULL && root->data <= prev->data)
        {
            return 0;
        }
        prev = root;
    }
}
```



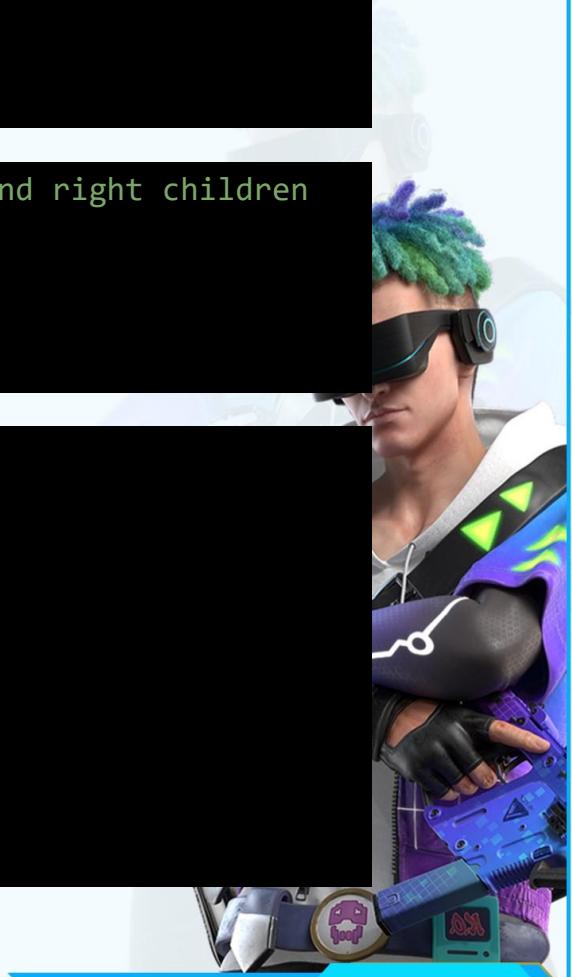
```
        return isBST(root->right);
    }
else
{
    return 1;
}
}
```

```
int main()
{
```

```
    // Constructing the root node -
Using Function (Recommended)
    struct node *p = createNode(5);
    struct node *p1 = createNode(3);
    struct node *p2 = createNode(6);
    struct node *p3 = createNode(1);
    struct node *p4 = createNode(4);
    // Finally The tree looks like this:
    //      5
    //      / \
    //     3   6
    //     / \
    //    1   4
```

```
    // Linking the root node with left and right children
    p->left = p1;
    p->right = p2;
    p1->left = p3;
    p1->right = p4;

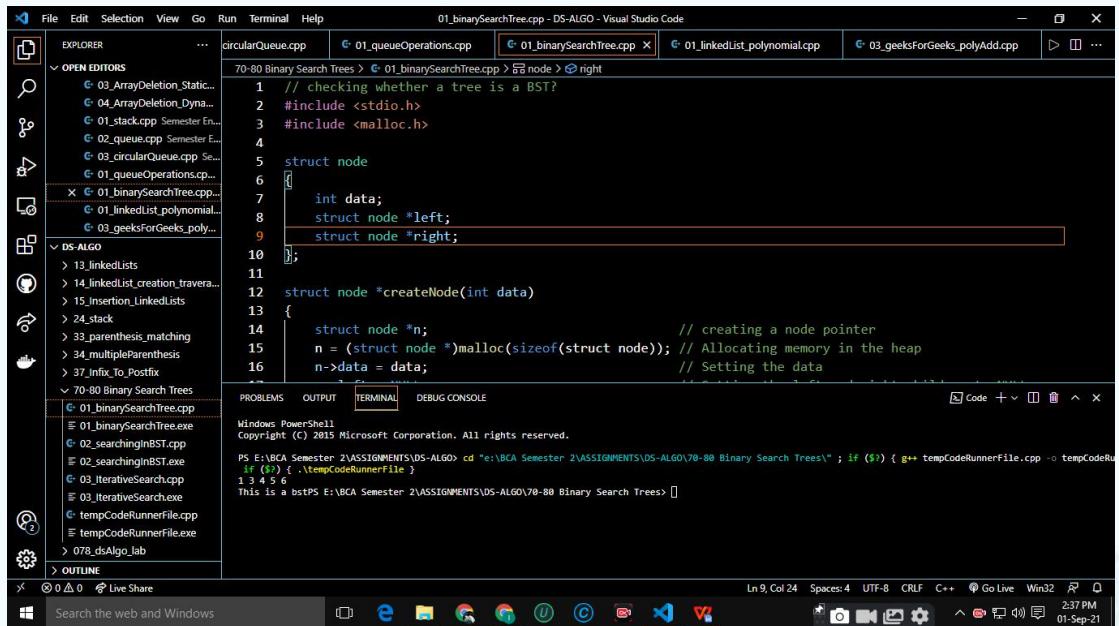
    // preOrder(p);
    // printf("\n");
    // postOrder(p);
    // printf("\n");
    inOrder(p);
    printf("\n");
    // printf("%d", isBST(p));
    if (isBST(p))
    {
        printf("This is a bst");
```



```

    }
else
{
    printf("This is not a bst");
}
return 0;
}

```

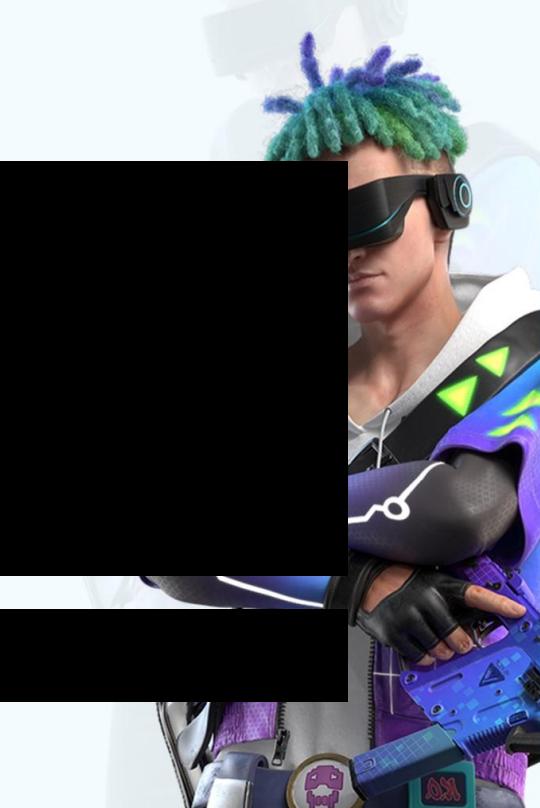


```

File Edit Selection View Go Run Terminal Help 01_binarySearchTree.cpp - DS-ALGO - Visual Studio Code
EXPLORER ... circularQueue.cpp 01_queueOperations.cpp 01_binarySearchTree.cpp X 01_linkedList_polynomial.cpp 03_geeksForGeeks_polyAdd.cpp
OPEN EDITORS 03_ArrayDeletion_Static... 04_ArrayDeletion_Dyna...
01_stack.cpp Semester E...
02_queue.cpp Semester E...
03_circularQueue.cpp Semester E...
01_queueOperations.cpp ...
01_binarySearchTree.cpp X 01_linkedList_polynomial...
03_geeksForGeeks_poly...
DS-ALGO ...
13.linkedLists ...
14.linkedList_creation_travers...
15.insertion_LinkedLists ...
24.stack ...
33_parenthesis_matching ...
34_multipleParenthesis ...
37_infix_To_Postfix ...
70-80 Binary Search Trees ...
01_binarySearchTree.cpp ...
01_binarySearchTree.exe ...
02_searchingInBST.cpp ...
02_searchingInBST.exe ...
03_iterativeSearch.cpp ...
03_iterativeSearch.exe ...
tempCodeRunnerFile.cpp ...
tempCodeRunnerFile.exe ...
078_dsAlgo_lab ...
OUTLINE ...
PROBLEMS ...
OUTPUT ...
DEBUG CONSOLE ...
Windows PowerShell ...
Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "e:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\70-80 Binary Search Trees"
$ ./01_binarySearchTree
3 5 6
This is a bst
PS E:\BCA Semester 2\ASSIGNMENTS\DS-ALGO\70-80 Binary Search Trees> []
Ln 9, Col 24 Spaces:4 UTF-8 CRLF C++ Go Live Win32 ...
237 PM 01-Sep-21

```

02_searchingInBST



```

#include <stdio.h>
#include <malloc.h>

struct node
{
    int data;
    struct node *left;
    struct node *right;
};

struct node *createNode(int data)
{

```

```

        struct node *n;                                // creating a node pointer
    ing a node pointer
        n = (struct node *)malloc(sizeof(struct node)); // Allocating memory in the heap
        n-
>data = data;                                     // Setting the
    data
        n-
>left = NULL;                                    // Setting the
    left and right children to NULL
        n-
>right = NULL;                                   // Setting the
    left and right children to NULL
        return n;                                     // Final
ly returning the created node
}

```

```

void preOrder(struct node *root)
{
    if (root != NULL)
    {
        printf("%d ", root->data);
        preOrder(root->left);
        preOrder(root->right);
    }
}

```

```

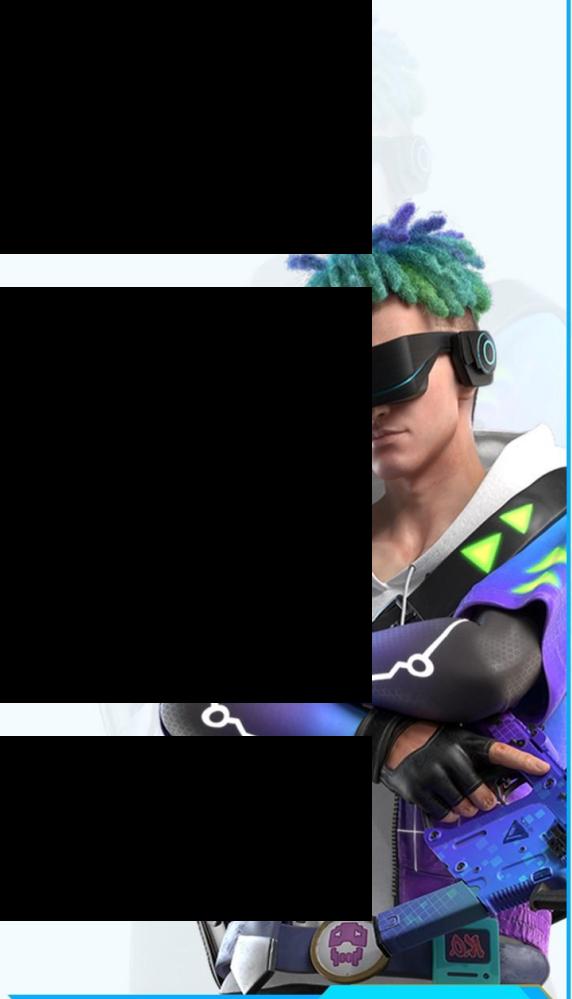
void postOrder(struct node *root)
{
    if (root != NULL)
    {
        postOrder(root->left);
        postOrder(root->right);
        printf("%d ", root->data);
    }
}

```

```

void inOrder(struct node *root)
{
    if (root != NULL)
    {

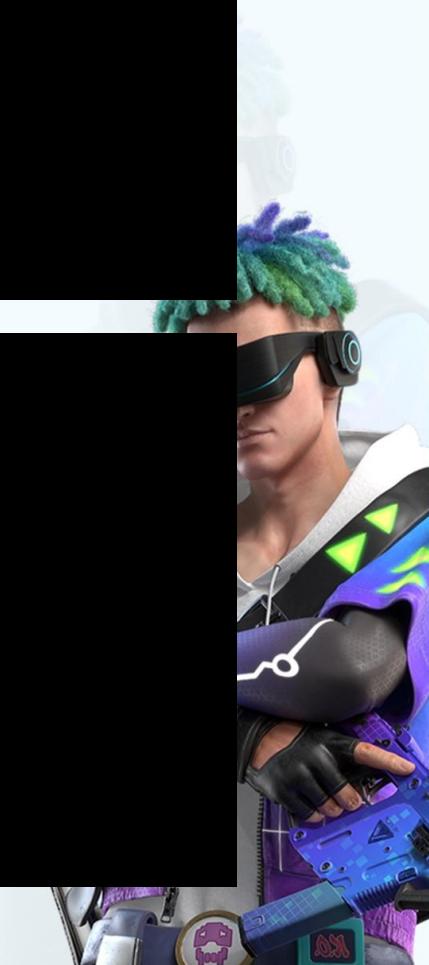
```



```
        inOrder(root->left);
        printf("%d ", root->data);
        inOrder(root->right);
    }
}
```

```
int isBST(struct node *root)
{
    static struct node *prev = NULL;
    if (root != NULL)
    {
        if (!isBST(root->left))
        {
            return 0;
        }
        if (prev != NULL && root->data <= prev->data)
        {
            return 0;
        }
        prev = root;
        return isBST(root->right);
    }
    else
    {
        return 1;
    }
}
```

```
struct node *search(struct node *root, int key)
{
    if (root == NULL)
    {
        return NULL;
    }
    if (key == root->data)
    {
        return root;
    }
    else if (key < root->data)
    {
```



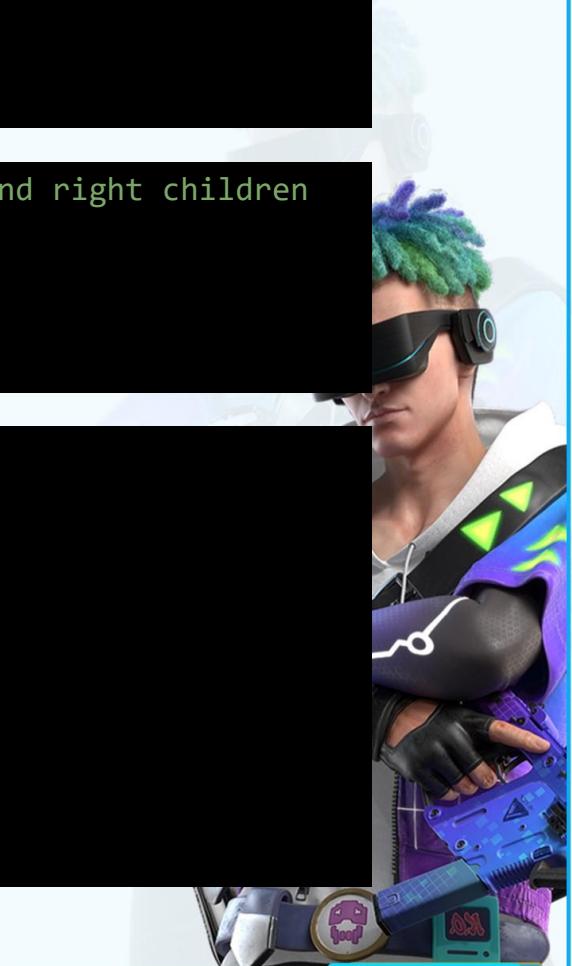
```
        return search(root->left, key);
    }
    else
    {
        return search(root->right, key);
    }
}
```

```
int main()
{
```

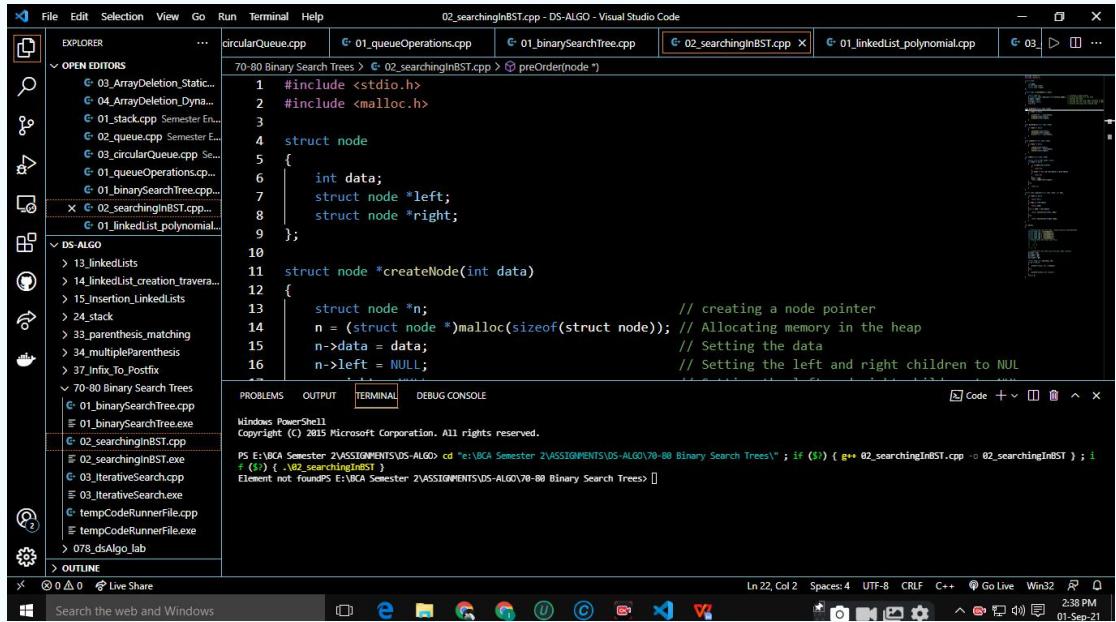
```
    // Constructing the root node -
    Using Function (Recommended)
    struct node *p = createNode(5);
    struct node *p1 = createNode(3);
    struct node *p2 = createNode(6);
    struct node *p3 = createNode(1);
    struct node *p4 = createNode(4);
    // Finally The tree looks like this:
    //      5
    //      / \
    //     3   6
    //     / \
    //    1   4
```

```
    // Linking the root node with left and right children
    p->left = p1;
    p->right = p2;
    p1->left = p3;
    p1->right = p4;
```

```
    struct node *n = search(p, 10);
    if (n != NULL)
    {
        printf("Found: %d", n->data);
    }
    else
    {
        printf("Element not found");
    }
    return 0;
```



{}



```

File Edit Selection View Go Run Terminal Help 02_searchingInBST.cpp - DS-ALGO - Visual Studio Code
EXPLORER 02_searchingInBST.cpp 01_binarySearchTree.cpp 02_searchingInBST.cpp 01_linkedList_polynomial.cpp 03...
OPEN EDITORS circularQueue.cpp 01_queueOperations.cpp 01_binarySearchTree.cpp 02_searchingInBST.cpp 01_linkedList...
70-80 Binary Search Trees > 02_searchingInBST.cpp > preOrder(node *)
1 #include <stdio.h>
2 #include <malloc.h>
3
4 struct node
5 {
6     int data;
7     struct node *left;
8     struct node *right;
9 };
10
11 struct node *createNode(int data)
12 {
13     struct node *n; // creating a node pointer
14     n = (struct node *)malloc(sizeof(struct node)); // Allocating memory in the heap
15     n->data = data; // Setting the data
16     n->left = NULL; // Setting the left and right children to NULL
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell Copyright (C) 2015 Microsoft Corporation. All rights reserved.
PS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO> cd "E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\70-80 Binary Search Trees\" ; g++ 02_searchingInBST.cpp -o 02_searchingInBST
f (*.*) { .\02_searchingInBST }
Element not foundPS E:\VCA Semester 2\ASSIGNMENTS\DS-ALGO\70-80 Binary Search Trees> []
Ln 22, Col 2 Spaces: 4 UTF-8 CR/LF C++ Go Live Win32 2:38 PM 01-Sep-21

```

05_TreeTraversal

```

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>

struct node
{
    int data;
    struct node *left;
    struct node *right;
};

struct node *temp;
int num;

struct node *create(int num)

```

```
{
```

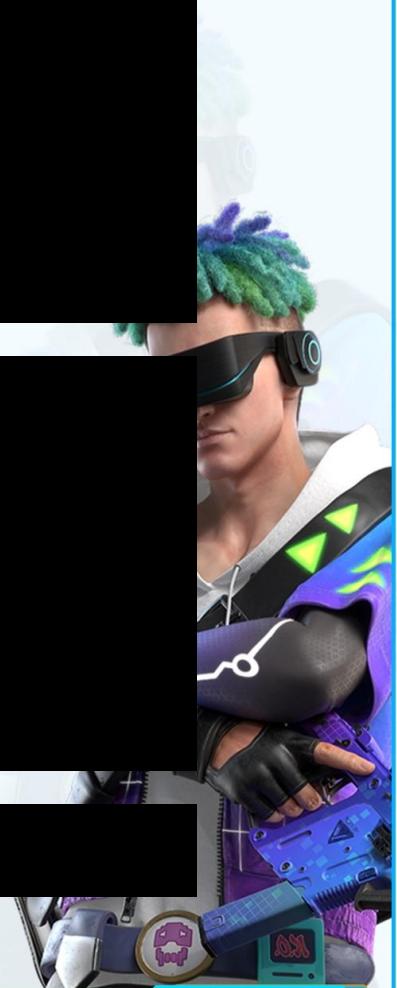
```
    temp = (struct node *)malloc(sizeof(struct node));
    if (temp == NULL)
        printf("not enough memory\n");
    else
    {
        temp->data = num;
        temp->left = NULL;
        temp->right = NULL;
    }
    return (temp);
}
```

```
struct node* insert(struct node *root, struct node *temp1)
{
    if (root ==NULL) return temp;
```

```
    if (temp1->data > root->data)
    {
        root->right=insert(root->right, temp1);
    }
    else
    {
        root->left=insert(root->left, temp1);
    }
    return root;
}
```

```
void preorder(struct node *root)
{
    if (root != NULL)
    {
        printf("%d\n", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
```

```
void inorder(struct node *root)
{
```



```
if (root != NULL)
{
    inorder(root->left);
    printf("%d\n", root->data);
    inorder(root->right);
}
```

```
void postorder(struct node *root)
{
    if (root != NULL)
    {
        postorder(root->left);
        postorder(root->right);
        printf("%d\n", root->data);
    }
}
```

```
struct node *minValueNode(struct node *root)
{
    struct node *current = root;
```

```
    while (current && current->left != NULL)
        current = current->left;

    return current;
}
```

```
struct node *deleteNode(struct node *root, int a)
{
    if (root == NULL)
        return root;

    if (a < root->data)
        root->left = deleteNode(root->left, a);

    else if (a > root->data)
        root->right = deleteNode(root->right, a);

    else
    {
        // node with only one child or no child
```



```

if (root->left == NULL)
{
    temp = root->right;
    free(root);
    return temp;
}
else if (root->right == NULL)
{
    temp = root->left;
    free(root);
    return temp;
}

// node with two children:
// Get the inorder successor
// (smallest in the right subtree)
temp = minValueNode(root->right);

// Copy the inorder
// successor's content to this node
root->data = temp->data;

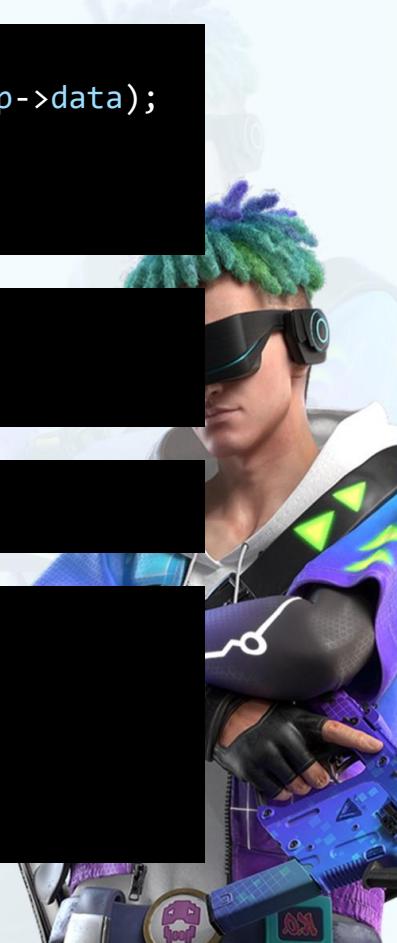
// Delete the inorder successor
root->right = deleteNode(root->right, temp->data);
}
return root;
}

int totalNodes(struct node *root)
{
    int count = 0;

    if (root == NULL)
        return 0;

    else
    {
        count+=1;
        count += totalNodes(root->left);
        count += totalNodes(root->right);
        return count;
    }
}

```



```

        }
    }

void deleteTree(struct node *root)
{
    if (root == NULL)
        return;

    /* first delete both subtrees */
    deleteTree(root->left);
    deleteTree(root->right);

    /* then delete the node */
    printf("\n Deleting node: %d", root->data);
    free(root);
}

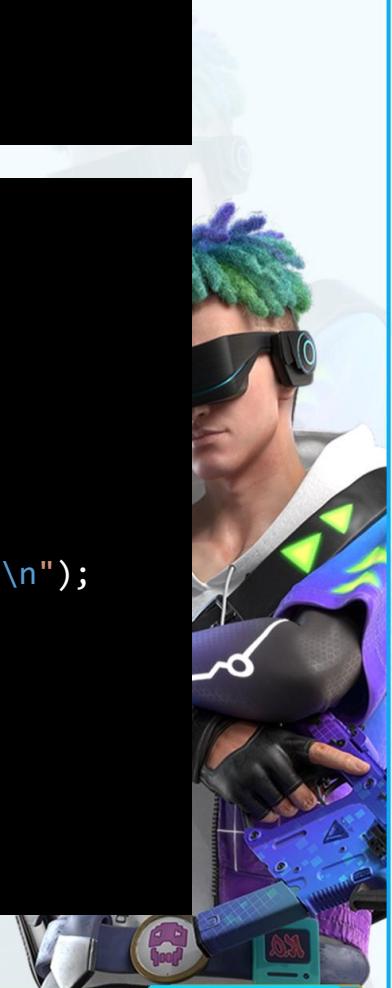
```

```

int main()
{
    int ch, val;
    struct node *root = NULL, *temp;

    do
    {
        printf("\n **MAIN MENU* \n");
        printf("1. create/Insert element\n");
        printf("2. Preorder Traversal\n");
        printf("3. Inorder Traversal\n");
        printf("4. Postorder Traversal\n");
        printf("5. Delete an element\n");
        printf("6. Count the total number of nodes\n");
        printf("7. Delete the tree\n");
        printf("8. Exit\n");
        printf("Enter your option : \n ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:

```



```

        printf("Enter the element\n");
        scanf("%d", &num);
        temp = create(num);
        root=insert(root, temp);
        break;

    case 2:
        printf("\n The elements of the tree are : \n");
        preorder(root);
        break;

    case 3:
        printf("\n The elements of the tree are : \n");
        inorder(root);
        break;

    case 4:
        printf("\n The elements of the tree are : \n");
        postorder(root);
        break;

    case 5:
        printf("\n Enter the element to be deleted : ");
        scanf("%d", &val);
        root = deleteNode(root, val);
        break;

    case 6:
        printf("\n Total no. of nodes = %d", totalNodes(
root));
        break;

    case 7:
        deleteTree(root);
        break;
    }

} while (ch != 8);

return 0;
}

```



A screenshot of Visual Studio Code showing an open project for "DS ALGO". The current file is "05_TreeTraversal.cpp". The code implements a binary search tree with functions for insertion, traversal, and search. The terminal tab shows a menu for tree operations. The status bar indicates the file has 10 lines, 3 spaces, and is saved in UTF-8.

```

File Edit Selection View Go Run Terminal Help
05_TreeTraversal.cpp - DS ALGO - Visual Studio Code
OPEN EDITORS
circularQueue.cpp | 01_queueOperations.cpp | 01_binarySearchTree.cpp | 02_searchingInBST.cpp | 05_TreeTraversal.cpp | 01_linkedLists.cpp ...
trees > 05_TreeTraversal.cpp > ...
1 #include <stdio.h>
2 #include <malloc.h>
3 #include <stdlib.h>
4
5 struct node
6 {
7     int data;
8     struct node *left;
9     struct node *right;
10 };
11
12 struct node *temp;
13 int num;
14
15 struct node *create(int num)
16 {
    ...
}
**MAIN MENU**
1. create/Insert element
2. Preorder Traversal
3. Inorder Traversal
4. Postorder Traversal
5. Delete an element
6. Count the total number of nodes
7. Delete the tree
8. Exit
Enter your option :
3
3

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Ln 10, Col 3 Spaces: 4 UTF-8 CRLF C++ Go Live Win32 2:40 PM 01-Sep-21

Rest of the lab practicals are at :

<https://drive.google.com/folderview?id=1pPRBJ3QjuW28oMx-enmI9ORcfdMZP4n1>