

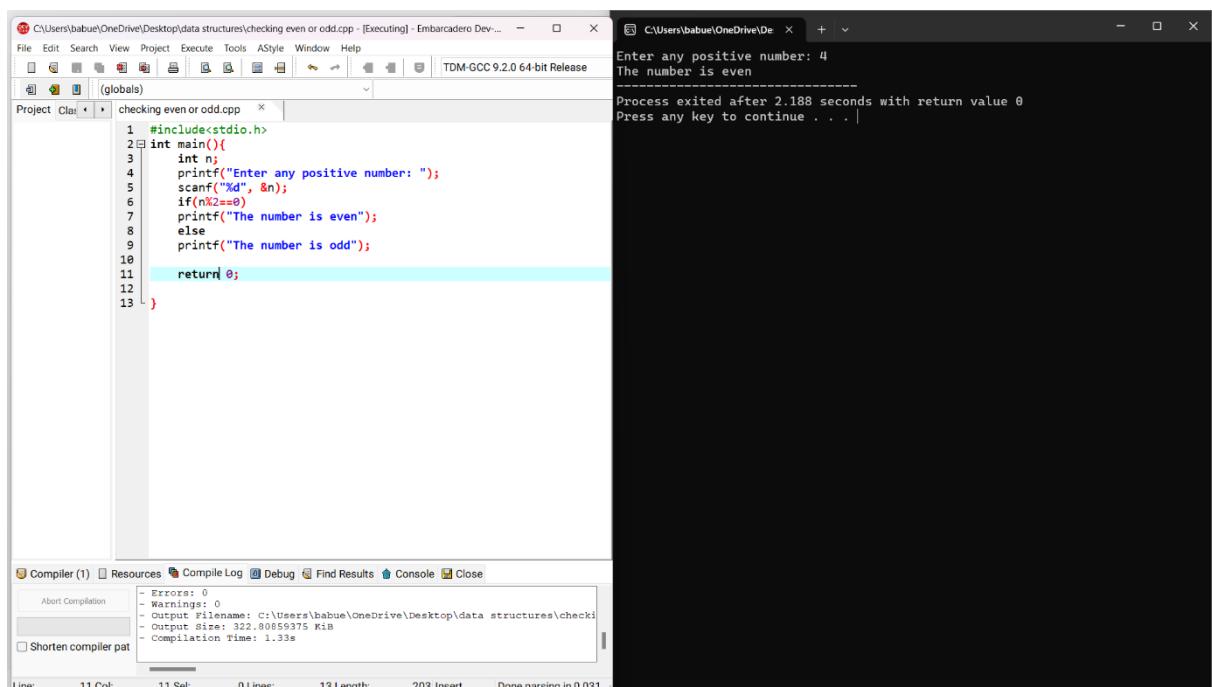
DAY – 1 DATA STRUCTURES PROGRAMS

- 1) Write a C program to find the given number is Even or Odd

Code:

```
#include<stdio.h>
int main(){
    int n;
    printf("Enter any positive number: ");
    scanf("%d", &n);
    if(n%2==0)
        printf("The number is even");
    else
        printf("The number is odd");

    return 0;
}
```



Sample input: 4

Output: The number is even.

- 2) Write a c program to find sum of first n numbers using any loop?

Code:

```
#include<stdio.h>
int main(){
    int n, i, sum = 0;
    printf("Enter any number: ");
    scanf("%d", &n);
    for(i = 1; i <= n; ++i){
        sum += i;
    }
    printf("The sum of first n numbers is : %d\n", sum);
    return 0;
}
```

The screenshot shows an IDE interface with two windows. The left window is the code editor, displaying the C program above. The right window is the terminal or output window, showing the execution results.

Code Editor (Left):

```
1 #include<stdio.h>
2 int main(){
3     int n, i, sum = 0;
4     printf("Enter any number: ");
5     scanf("%d", &n);
6     for(i = 1; i <= n; ++i){
7         sum += i;
8     }
9     printf("The sum of first n numbers is : %d\n", sum);
10    return 0;
11 }
```

Output Window (Right):

```
Enter any number: 5
The sum of first n numbers is : 15
-----
Process exited after 2.28 seconds with return value 0
Press any key to continue . . .
```

Sample input = 5

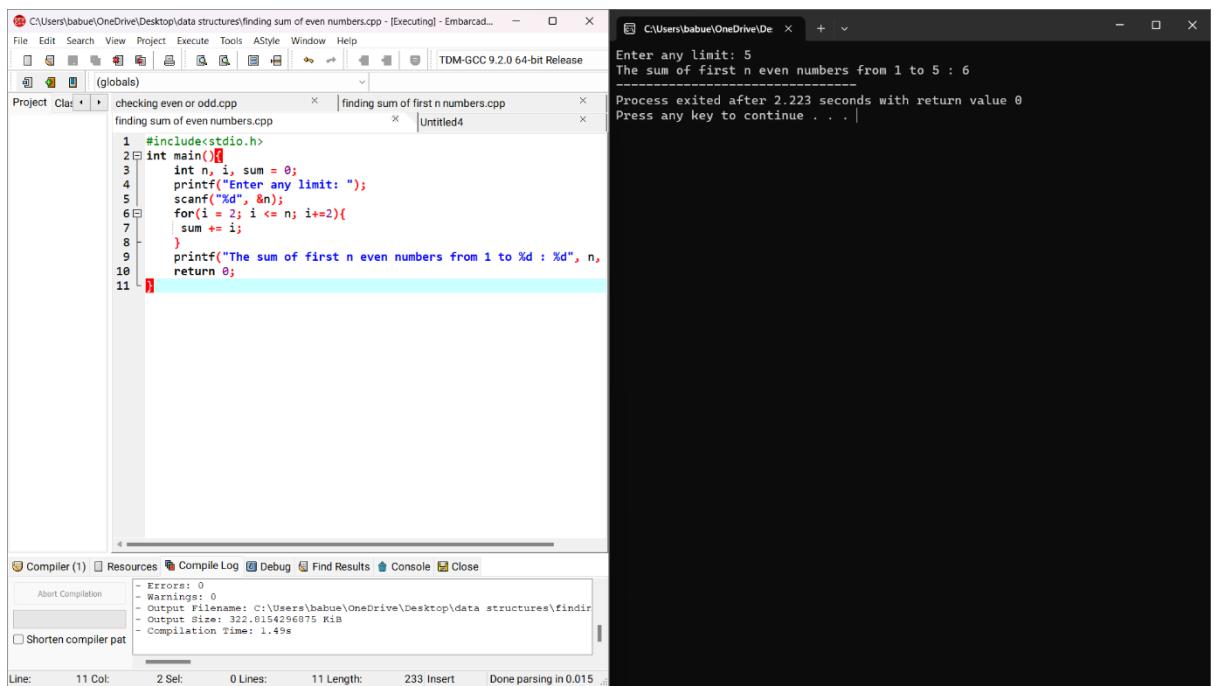
Output : sum = 15

- 3) Write a C program to find sum of even numbers with first n numbers?

Code:

```
#include<stdio.h>

int main(){
    int n, i, sum = 0;
    printf("Enter any limit: ");
    scanf("%d", &n);
    for(i = 2; i <= n; i+=2){
        sum += i;
    }
    printf("The sum of first n even numbers from 1 to %d : %d", n, sum);
    return 0;
}
```



Sample input : 5

Output: The sum of first n even numbers from 1 to 5: 6

- 4) Write a C program to find sum of odd numbers the first n numbers?

Code:

```
#include<stdio.h>

int main(){
    int n, i, sum = 0;
    printf("Enter the limit: ");
    scanf("%d", &n);
    for(i = 1; i <=n; i++){
        if(i % 2 != 0){
            printf("%d\n", i);
            sum = sum + i;
        }
    }
    printf("The sum of Odd numbers from 0 to %d is: %d", n, sum);
    return 0;
}
```

The screenshot shows the Embarcadero IDE interface. On the left, the code editor displays the C program. On the right, the terminal window shows the execution results. The terminal output is as follows:

```
Enter the limit: 5
1
3
5
The sum of Odd numbers from 0 to 5 is: 9
Process exited after 4.119 seconds with return value 0
Press any key to continue . . .
```

Sample input: 5 and Output: 9

- 5) Write a C program to find the factorial of a given number with Recursion?

Code:

```
#include<stdio.h>

int fact(int n){

    if(n == 1){

        return n;

    }

    return n*fact(n-1);

}

int main(){

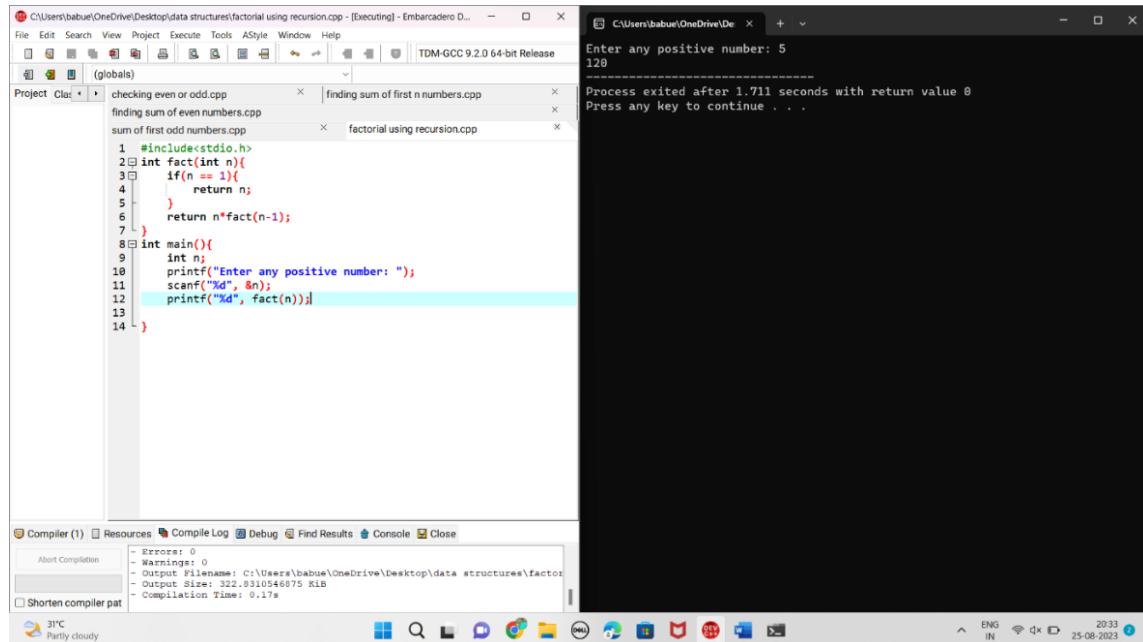
    int n;

    printf("Enter any positive number: ");

    scanf("%d", &n);

    printf("%d", fact(n));

}
```



Sample input: 5

Output : 120

- 6) Write a C program to find the factorial of a given number without Recursion?

Code:

```
#include<stdio.h>

int main()
{
    int n,i,fact=1;
    printf("Enter any number: ");
    scanf("%d", &n);
    for(i = 1; i <= n; i++){
        fact = fact * i;
    }
    printf("The factorial of %d is : %d", n, fact);
    return 0;
}
```

The screenshot shows the Embarcadero Dev-C++ IDE interface. On the left, the code editor displays a C program named 'factorial.cpp' with line numbers 1 through 23. The code implements a loop to calculate the factorial of a user-specified number. On the right, the terminal window shows the execution of the program. It prompts the user to enter a number (5), displays the result (The factorial of 5 is : 120), and then exits. Below the terminal is the compiler output window, which shows a successful compilation with no errors or warnings.

```
C:\Users\babue\OneDrive\Desktop\data structures\factorial.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools ASStyle Window Help
TDM-GCC 9.2.0 64-bit Release
Project Class factorial using recursion.cpp factorial.cpp
1 #include<stdio.h>
2 int main()
3 {
4
5     int n,i,fact=1;
6     printf("Enter any number: ");
7     scanf("%d", &n);
8     for(i = 1; i <= n; i++){
9         fact = fact * i;
10    }
11    printf("The factorial of %d is : %d", n, fact);
12    return 0;
13 }

Compiler (2) Resources Compile Log Debug Find Results Console Close
Abort Compilation
Errors: 0
Warnings: 0
Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\factorial
Output Size: 322.7841796875 KiB
Compilation Time: 0.17s
Shorten compiler path
Line: 13 Col: 13 Sel: 0 Lines: 23 Length: 241 Insert Done parsing in 0.015s
```

Sample input :5

Output: 120

7) Write a C program to find or to generate Fibonacci series with Recursion?

Code:

```
#include<stdio.h>
int main(){
    int n1 = 0, n2 = 1, n3, i, number;
    printf("Enter the number of elements : ");
    scanf("%d", &number);
    printf("\n%d%d", n1,n2);
    for(i = 2; i < number; ++i){
        n3 = n1 + n2;
        printf("%d", n3);
        n1 = n2;
        n2 = n3;
    }
    return 0;
}
```

The screenshot shows the Embarcadero Dev-C++ IDE interface. On the left, the code editor displays the C program for generating a Fibonacci series using recursion. On the right, the terminal window shows the output of the program. The user enters '15' as the number of elements. The program then prints the Fibonacci sequence from 0 to 144, followed by a message indicating the process exited after 5.005 seconds.

```
C:\Users\babue\OneDrive\Desktop\data structures\fibonacci series.cpp - [Executing] - Embarcadero Dev-C++ 6.0
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
factorial using recursion.cpp factorial.cpp fibonacci series.cpp
Project Class + Enter the number of elements : 15
1 #include<stdio.h>
2 int main(){
3     int n1 = 0, n2 = 1, n3, i, number;
4     printf("Enter the number of elements : ");
5     scanf("%d", &number);
6     printf("\n%d%d", n1,n2);
7     for(i = 2; i < number; ++i){
8         n3 = n1 + n2;
9         printf("%d", n3);
10        n1 = n2;
11        n2 = n3;
12    }
13    return 0;
14 }

Process exited after 5.005 seconds with return value 0
Press any key to continue . . .
```

Sample input: 15

Output : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377.

8)Write a C program to generate Fibonacci series with recursion?

Code:

```
#include<stdio.h>

int fi(int n){

    if(n == 0){
        return 0;
    }

    else if(n == 1){
        return 1;
    }

    else{
        return(fi(n-1)+fi(n-2));
    }
}

int main(){

    int fi(int n);
    int n, i = 0, c;
    scanf("%d", &n);
    printf("fibonacci series \n");
    for(int c=1;c<=n;c++){
        printf("%d", fi(i));
        i++;
    }
    return 0;
}
```

```
15
fibonacci series
01123581321345589144233377
Process exited after 10.33 seconds with return value 0
Press any key to continue . . .

1 #include<stdio.h>
2 int fi(int n){
3     if(n == 0){
4         return 0;
5     }
6     else if(n == 1){
7         return 1;
8     }
9     else{
10         return(fi(n-1)+fi(n-2));
11     }
12 }
13 int main(){
14     int fi(int n);
15     int i = 0, c;
16     scanf("%d", &n);
17     printf("fibonacci series \n");
18     for(int c=1;c<=n;c++){
19         printf("%d", fi(i));
20         i++;
21     }
22     return 0;
23 }
```

Compiler (2) Resources Compile Log Debug Find Results Console Close

Abort Compilation - Errors: 0
- Warnings: 0
- Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\fibona
- Output Size: 323.0068359375 KB
- Compilation Time: 0.17s

Line: 22 Col: 14 Sel: 0 Lines: 23 Length: 320 Insert Done parsing in 0 sec ..

Sample input : 15

Output: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377.

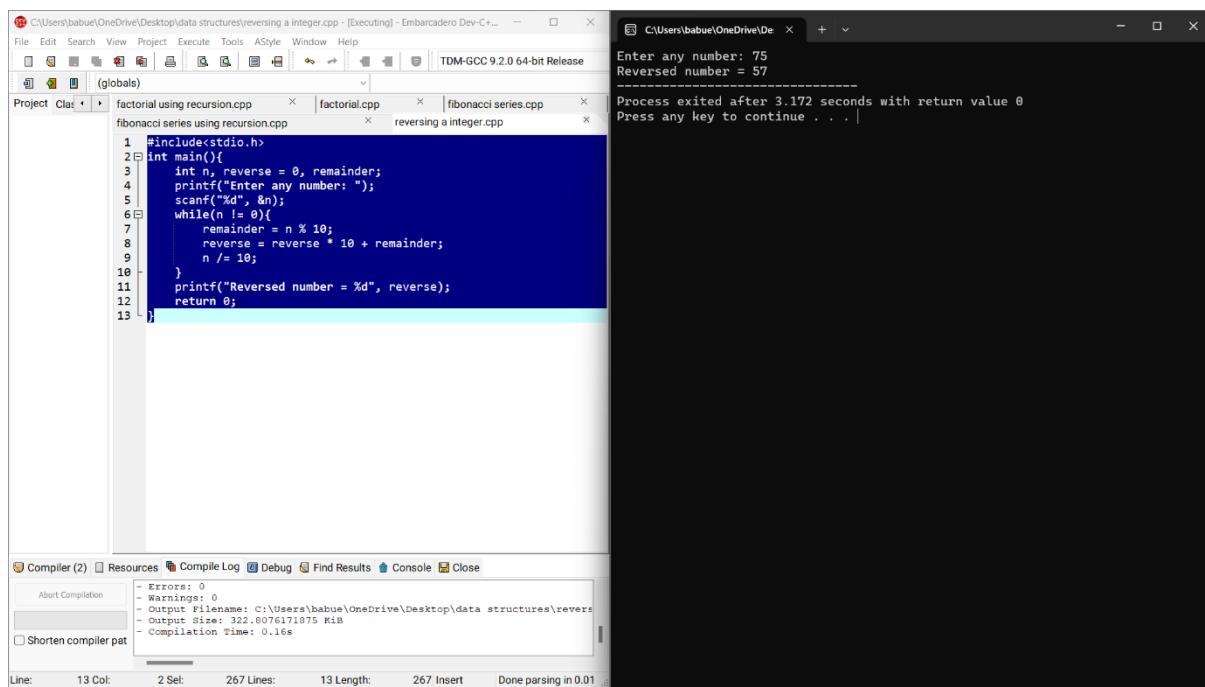
9) Write a C program to reverse a given number?

Code:

```
#include<stdio.h>

int main(){

    int n, reverse = 0, remainder;
    printf("Enter any number: ");
    scanf("%d", &n);
    while(n != 0){
        remainder = n % 10;
        reverse = reverse * 10 + remainder;
        n /= 10;
    }
    printf("Reversed number = %d", reverse);
    return 0;
}
```



Sample input: 75

Output: 57

10) Write a C program to check palindrome or not?

Code:

```
#include<stdio.h>

int main()
{
    int n, r, sum = 0, temp;
    printf("Enter the number: ");
    scanf("%d", &n);
    temp = n;
    while(n>0){
        r = n%10;
        sum = (sum*10)+r;
        n = n/10;
    }
    if(temp==sum)
        printf("Palindrome number");
    else
        printf("not palindrome");
    return 0;
}
```

The screenshot shows the Embarcadero Dev C++ IDE interface. On the left, the Project Explorer lists several files: factorial using recursion.cpp, factorial.cpp, fibonacci series.cpp, reversing a integer.cpp, and palindrome checking.cpp. The palindrome checking.cpp file is open in the main editor window, displaying the following C code:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n, r, sum = 0, temp;
5     printf("Enter the number: ");
6     scanf("%d", &n);
7     temp = n;
8     while(n>0){
9         r = n%10;
10        sum = (sum*10)+r;
11        n = n/10;
12    }
13    if(temp==sum)
14        printf("Palindrome number");
15    else
16        printf("not palindrome");
17    return 0;
18 }
```

Below the editor is the Compiler (2) tab of the status bar, which shows compilation results:

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\palindrome checking
- Output Size: 322.0076171875 Kib
- Compilation Time: 0.16s

The status bar also displays current input values: Line: 18 Col: 18 Sel: 284 Lines: 18 Length: 284 Insert.

In the bottom right corner of the IDE window, there is a terminal window showing the execution output:

```
Enter the number: 575
Palindrome number
Process exited after 4.004 seconds with return value 0
Press any key to continue . . .
```

Sample input:575

Output: It is palindrome number.

11) Write a C program to check the given Armstrong or not?

Code:

```
#include<stdio.h>

int main()
{
    int num, originalnum, rem, res=0;
    printf("Enter a three digit number: ");
    scanf("%d", &num);
    originalnum = num;
    while(originalnum != 0){
        rem = originalnum % 10;
        res += rem * rem * rem;
        originalnum /= 10;
    }
    if(res == num)
        printf("%d is an armstrong number", num);
    else
        printf("%d is not an armstrong number", num);
    return 0;
}
```

The screenshot shows the Embarcadero C++Builder IDE interface. The main window displays a code editor with a C++ file named "armstrong number chkking.cpp". The code implements a program to check if a three-digit number is an Armstrong number. The terminal window on the right shows the execution results for the input 153. Below the code editor is a Compiler window showing build statistics.

```
1 #include<stdio.h>
2 int main()
3 {
4     int num, originalnum, rem, res=0;
5     printf("Enter a three digit number: ");
6     scanf("%d", &num);
7     originalnum = num;
8     while(originalnum != 0){
9         rem = originalnum % 10;
10        res += rem * rem * rem;
11        originalnum /= 10;
12    }
13    if(res == num)
14        printf("%d is an Armstrong number", num);
15    else
16        printf("%d is not an Armstrong number", num);
17    return 0;
18 }
```

Compiler (2) Resources Compile Log Debug Find Results Console Close

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\armstrong number chkking.cpp
- Output Size: 322.8125 KB
- Compilation Time: 0.14s

Line: 18 Col: 2 Sel: 390 Lines: 18 Length: 390 Insert Done parsing in 0.01

Sample input: 153

Output: It is Armstrong number.

- 1) Write a C program to initialise the array Dynamically and print the array?

Code:

```
#include<stdio.h>

int main(){

    int n;

    printf("enter the size of array:");

    scanf("%d",&n);

    printf("Enter the elements:");

    int a[n];

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

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

    }

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

        printf("%d ",a[i]);

    }

    return 0;
}
```

The screenshot shows the OnlineGDB beta interface. The code editor contains the provided C program. The browser's address bar shows 'onlinegdb.com'. The developer tools' console tab displays the program's output: 'enter the size of array:', 'Enter the elements:', followed by the input '1 2 3 4 5', and finally '...Program finished with exit code 0'. A sidebar on the left offers various services like IDE, Classroom, and Programming Questions.

```
1 //*****
2 //***** Welcome to GDB Online.
3 //***** GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,
4 //***** C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.
5 //***** Code, Compile, Run and Debug online from anywhere in world.
6
7
8 *****
9 #include<stdio.h>
10 int main(){
11     int n;
12     printf("enter the size of array:");
13     scanf("%d",&n);
14     printf("Enter the elements:");
15     int a[n];
16     for(int i=0;i<n;i++){
17         scanf("%d",&a[i]);
18     }
19     for(int i=0;i<n;i++){
20         printf("%d ",a[i]);
21     }
22 }
23 }
```

input

```
enter the size of array:
Enter the elements:
1 2 3 4 5
...Program finished with exit code 0
Press ENTER to exit console.
```

Sample input: 5 and output: 1,2,3,4,5

2) Write a C program to find sum of elements in a given array?

Code:

```
#include<stdio.h>

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

The screenshot shows the Embarcadero Dev-C++ 6.3 IDE interface. The left pane displays the code editor with the file 'printing the array dynamically.cpp' open. The right pane shows the terminal window where the program is being run. The terminal output is as follows:

```
C:\Users\babue\OneDrive\Desktop\data structures\sum of elements in an array.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools Style Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Class... printing the array dynamically.cpp Untitled2.cpp sum of elements in an array.cpp
C:\Users\babue\OneDrive\Desktop\data structures\sum of elements in an array.cpp
1 #include<stdio.h>
2 int main(){
3     int size;
4     printf("Enter the size of the array: ");
5     scanf("%d", &size);
6     int arr[size];
7     printf("Enter the elements of the array: ");
8     for(int i = 0; i < size; i++){
9         scanf("%d", &arr[i]);
10    }
11    int sum = 0;
12    for(int i = 0; i < size; i++){
13        sum += arr[i];
14    }
15    printf("sum of array elements : %d\n", sum);
16    return 0;
17 }

C:\Users\babue\OneDrive\Desktop\data structures\sum of elements in an array.cpp
Enter the size of the array: 5
Enter the elements of the array: 2 3 4 5 6
sum of array elements : 20
Process exited after 18.04 seconds with return value 0
Press any key to continue . . . |
```

The bottom status bar indicates the following compilation information:

- Compiler (4)
- Resources
- Compile Log
- Debug
- Find Results
- Console
- Close

Output:
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\sum of elements in an array.exe
- Output Size: 323.3154296875 KiB
- Compilation Time: 0.28s

Line: 3 Col: 14 Sel: 0 Lines: 17 Length: 362 Insert Done parsing in 0.032 seconds

3) Write a c program to find sum of even and sum of odd numbers in the given array?

Code:

```
#include<stdio.h>

int main(){
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter the elements of the array: ");
    for(int i = 0; i < size; i++){
        scanf("%d", &arr[i]);
    }
    int sum_even = 0, sum_odd = 0;
    for(int i = 0; i < size; i++){
        if(arr[i] % 2 == 0){
            sum_even += arr[i];
        }
        else{
            sum_odd += arr[i];
        }
    }
    printf("Sum of even numbers: %d\n", sum_even);
    printf("Sum of odd numbers: %d\n", sum_odd);
    return 0;
}
```

The screenshot shows a C++ development environment with the following components:

- Code Editor:** Displays the source code for a program named "printing the array dynamically.cpp". The code reads an array size and elements from the user, then calculates and prints the sum of even and odd numbers.
- Compiler Output:** Shows the compilation results for "printing the array dynamically.cpp". It includes statistics like errors, warnings, output filename, size, and compilation time.
- Console:** Displays the program's runtime output. It asks for the array size (5), lists the array elements (2 4 5 6 3), and then prints the sums of even (12) and odd (8) numbers.

```
#include<stdio.h>
int main(){
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter the elements of the array: ");
    for(int i = 0; i < size; i++){
        scanf("%d", &arr[i]);
    }
    int sum_even = 0, sum_odd = 0;
    for(int i = 0; i < size; i++){
        if(arr[i] % 2 == 0){
            sum_even += arr[i];
        }
        else{
            sum_odd += arr[i];
        }
    }
    printf("Sum of even numbers: %d\n", sum_even);
    printf("Sum of odd numbers: %d\n", sum_odd);
    return 0;
}
```

Compiler (4) Resources Compile Log Debug Find Results Console Close

Abort Compilation - Errors: 0
- Warnings: 0
- Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\sum of even and odd numbers in an array.cpp
- Output Size: 323.3271484375 Kib
- Compilation Time: 0.17s

Line: 22 Col: 14 Sel: 0 Lines: 23 Length: 499 Insert Done parsing in 0 sec

4) Write a c program to perform the following operations

a) Insert an element in an array

Code:

```
#include<stdio.h>
int main(){
    int n,k,p,i;
    printf("enter the size of array:");
    scanf("%d",&n);
    printf("Enter the elements:");
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Enter the element to insert:");
    scanf("%d",&k);
    printf("Enter the element position to insert:");
    scanf("%d",&p);
    for(i=n;i>=p;i--){
        a[i]=a[i-1];
    }
    a[p]=k;
    n++;
    for(int i=0;i<n;i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

The screenshot shows the OnlineGDB beta IDE interface. The code editor contains the C program provided above. The input window shows the user entering the size of the array (2), the elements (3, 4, 5, 6), the element to insert (7), and its position (2). The output window displays the final array [2, 3, 7, 4, 5, 6].

```
#include<stdio.h>
int main(){
    int n,k,p,i;
    printf("enter the size of array:");
    scanf("%d",&n);
    printf("Enter the elements:");
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Enter the element to insert:");
    scanf("%d",&k);
    printf("Enter the element position to insert:");
    scanf("%d",&p);
    for(i=n;i>=p;i--){
        a[i]=a[i-1];
    }
    a[p]=k;
    n++;
    for(int i=0;i<n;i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

Have fun taking surveys
and get paid!
GOT AN OPINION?
SHARE AND GET REWARDED!
Okutena API

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

b)Delete an element

Code:

```
#include<stdio.h>

int main(){
    int n,k,p,i;
    printf("enter the size of array:");
    scanf("%d",&n);
    printf("Enter the elements:");
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Enter the element position to delete:");
    scanf("%d",&p);
    for(i=0;i<n;i++){
        if(i==p){
            for(int j=i;j<n;j++){
                a[j]=a[j+1];
            }
        }
    }
    for(int i=0;i<n-1;i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

Screenshot of the OnlineGDB beta IDE interface showing a C program to print an array.

The code in main.c is:

```
#include<stdio.h>
int main(){
    int n,k,p,i;
    printf("Enter the size of array:");
    scanf("%d",&n);
    printf("Enter the elements:");
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    printf("Enter the element position to delete:");
    scanf("%d",&p);
    for(i=0;i<n;i++){
        if(i==p){
            for(int j=i;j<n;j++)
                a[j]=a[j+1];
        }
    }
    for(int i=0;i<n-1;i++){
        printf("%d ",a[i]);
    }
    return 0;
}
```

The input provided is:

```
Enter the size of array:5
Enter the elements:2
3
4
5
6
Enter the element position to delete:3
2 3 4 6
```

The output is:

```
...Program finished with exit code 0
Press ENTER to exit console.
```

- 5) Write a c program to merge two arrays?

Code:

```
#include<stdio.h>
int main(){
int n,m,i,j,o=0;
printf("enter the size of array 1:");
scanf("%d",&n);
printf("Enter the elements:");
int a[n];
for(i=0;i<n;i++){
scanf("%d",&a[i]);
}
printf("enter the size of array 2:");
scanf("%d",&m);
printf("Enter the elements:");
int b[m],k[m+n];
for(j=0;j<m;j++){
scanf("%d",&b[j]);
}
for(i=0;i<n;i++){
k[i]=a[i];
}
j=i;
for(i=0;i<m;i++){
k[j]=b[i];
j++;
}

for(int i=0;i<m+n;i++){
printf("%d ",k[i]);
}
return 0;
}
```

Program to Print Array in C - Scs | Online C Compiler | GDB online Debugger | Compile

onlinegdb.com

Gmail YouTube Maps News Translate

Language: C

OnlineGDB beta
online compiler and debugger for c/c++

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

GOT AN OPINION?
SHARE AND GET REWARDED
Orkutun API

Have fun taking surveys
and get paid!

```
main.c
1 #include<stdio.h>
2 int main(){
3     int n,m,i,j,o=0;
4     printf("enter the size of array 1:");
5     scanf("%d",&n);
6     printf("Enter the elements:");
7     int a[n];
8     for(i=0;i<n;i++){
9         scanf("%d",&a[i]);
10    }
11    printf("enter the size of array 2:");
12    scanf("%d",&m);
13    printf("Enter the elements:");
14    int b[m],k[m*n];
15    for(j=0;j<m;j++){
16        scanf("%d",&b[j]);
17    }
18    for(i=0;i<n;i++){
19        k[i]=a[i];
20    }
21    j=i;
22    for(i=0;i<m;i++){

```

input

enter the size of array 1:3
Enter the elements:1
2
3
enter the size of array 2:3
Enter the elements:4
5
6
1 2 3 4 5 6
...Program finished with exit code 0
Press ENTER to exit console.

6) Write a c program to find duplicate value in array?

Code:

```
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int i, j, size, count = 0;
    printf("Enter size of the array : ");
    scanf("%d", &size);
    printf("Enter elements in array : ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    for(i=0; i<size; i++)
    {
        for(j=i+1; j<size; j++)
        {
            if(arr[i] == arr[j])
            {
                count++;
                break;
            }
        }
    }
    printf("\nTotal number of duplicate elements found in array = %d", count);
    return 0;
}
```

C program count total duplicate | GDB online Debugger | Compiler | onlinergdb.com

Gmail YouTube Maps News Translate Language C

OnlineGDB beta
online compiler and debugger for c/c++
code, compile, run, debug, share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

GOT AN OPINION?
SHARE AND SEE REWARDS
@RokutenAIP

Have fun taking surveys
and get paid!

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

```
main.c
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int main()
4 {
5     int arr[MAX_SIZE];
6     int i, j, size, count = 0;
7     printf("Enter size of the array : ");
8     scanf("%d", &size);
9     printf("Enter elements in array : ");
10    for(i=0; i<size; i++)
11    {
12        scanf("%d", &arr[i]);
13    }
14    for(i=0; i<size; i++)
15    {
16        for(j=i+1; j<size; j++)
17        {
18            if(arr[i] == arr[j])
19            {
20                count++;
21                break;
22            }
23        }
24    }
25    printf("\nTotal number of duplicate elements found in array = %d", count);
26    return 0;
27 }
28 }
```

Enter size of the array : 5
Enter elements in array : 1
2
3
4
Total number of duplicate elements found in array = 1

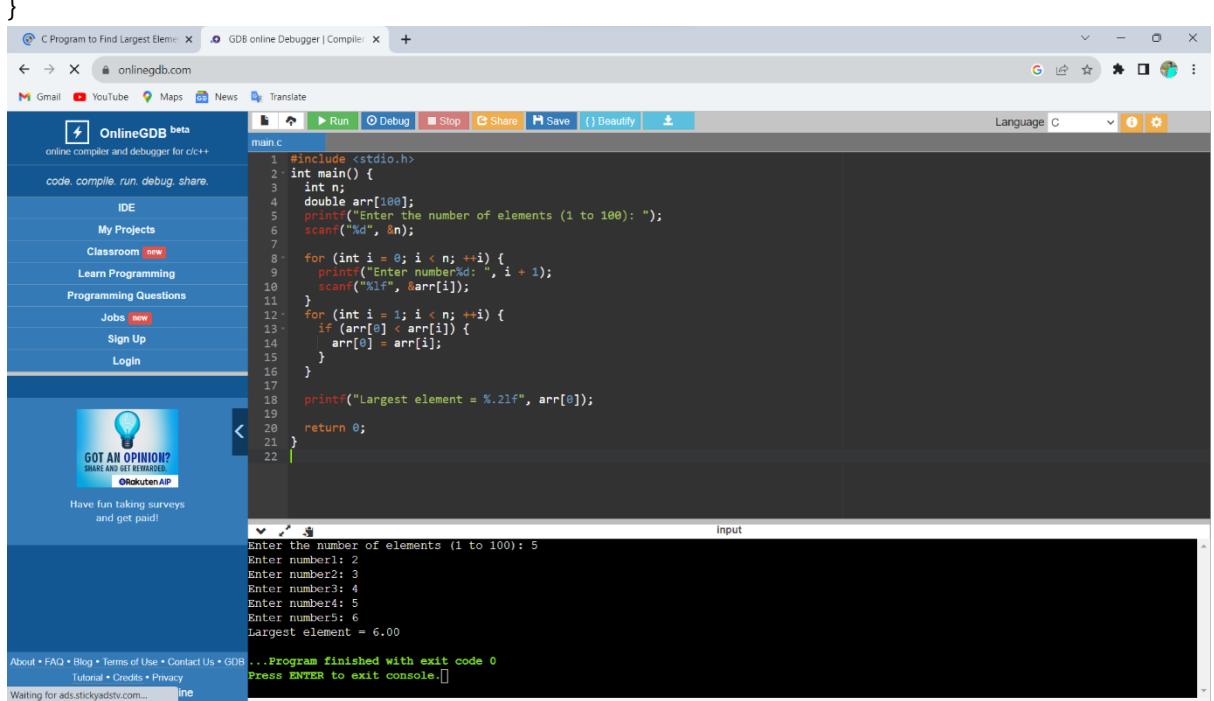
- 7) Write a C program to find greatest element among array elements?

Code:

```
#include <stdio.h>
int main() {
    int n;
    double arr[100];
    printf("Enter the number of elements (1 to 100): ");
    scanf("%d", &n);

    for (int i = 0; i < n; ++i) {
        printf("Enter number%d: ", i + 1);
        scanf("%lf", &arr[i]);
    }
    for (int i = 1; i < n; ++i) {
        if (arr[0] < arr[i]) {
            arr[0] = arr[i];
        }
    }
    printf("Largest element = %.2lf", arr[0]);
}
```

```
return 0;
}
```



- 8) Write a C program to search an element in array using linear search?

Code:

```
#include <stdio.h>

int main()
{
    int array[100], search, c, number;

    printf("Enter the number of elements in array\n");
    scanf("%d",&number);

    printf("Enter %d numbers\n", number);
    for ( c = 0 ; c < number ; c++ )
        scanf("%d",&array[c]);

    printf("Enter the number to search\n");
    scanf("%d",&search);

    for ( c = 0 ; c < number ; c++ )
    {
        if ( array[c] == search )

        {
            printf("%d is present at location %d.\n", search, c+1);
            break;
        }
    }

    if ( c == number )
        printf("%d is not present in array.\n", search);

    return 0;
}
```

C Program to Find an Element Using Linear Search

GDB online Debugger | Compiler

onlinegdb.com

Translate

Language: C

IDE My Projects Classroom Learn Programming Programming Questions Jobs Sign Up Login

GOT AN OPINION? SHARE AND GET REWARDED! Orkutun API

Have fun taking surveys and get paid!

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

```
main.c
1 #include <stdio.h>
2 int main()
3 {
4     int array[100], search, c, number;
5     printf("Enter the number of elements in array\n");
6     scanf("%d",&number);
7     printf("Enter %d numbers\n", number);
8     for ( c = 0 ; c < number ; c++ )
9         scanf("%d",&array[c]);
10    printf("Enter the number to search\n");
11    scanf("%d",&search);
12    for ( c = 0 ; c < number ; c++ )
13    {
14        if ( array[c] == search ) |
15        {
16            printf("%d is present at location %d.\n", search, c+1);
17            break;
18        }
19        if ( c == number )
20            printf("%d is not present in array.\n", search);
21    }
22    return 0;
23 }
```

Input

```
Enter the number of elements in array
5
Enter 5 numbers
22
33
44
55
66
Enter the number to search
33
33 is present at location 2.
```

- 9) Write a c program to search an element in array using binary search?

Code:

```
//Program Name: BinarySearch.c #include<stdio.h>
int main()
{
    int c, first, last, middle, n, search, array[100];
    printf("Enter number of elements\n");
    scanf("%d",&n);
    printf("Enter %d integers\n", n);
    for ( c = 0 ; c < n ; c++ )
        scanf("%d",&array[c]);
    printf("Enter value to find\n");
    scanf("%d",&search);
    first = 0;
    last = n - 1;
    middle = (first+last)/2;
    while( first <= last )
    {
        if ( array[middle] < search )
            first = middle + 1;
        else if ( array[middle] == search )
        {
            printf("%d found at location %d.\n", search, middle+1);
            break;
        }
        else
            last = middle - 1;
        middle = (first + last)/2;
    }
    if ( first > last )
        printf("Not found! %d is not present in the list.\n", search);
}
```

```
return 0;  
}
```

The screenshot shows a web browser window for the OnlineGDB website. The URL is onlinegdb.com. The page title is "C Program to find an Element using Binary Search | GDB online Debugger | Compiler". The code editor contains a C program named "main.c" which implements a binary search algorithm. The code is as follows:

```
#include<stdio.h>
int main()
{
    int c, first, last, middle, n, search, array[100];
    printf("Enter number of elements\n");
    scanf("%d",&n);
    printf("Enter %d integers\n", n);
    for ( c = 0 ; c < n ; c++)
        scanf("%d",&array[c]);
    printf("Enter value to find\n");
    scanf("%d",&search);
    first = 0;
    last = n - 1;
    middle = (first+last)/2;
    while( first <= last )
    {
        if ( array[middle] < search )
            first = middle + 1;
        else if ( array[middle] == search )
        {
            printf("%d found at location %d.\n", search, middle+1);
            break;
        }
        else
            last = middle - 1;
    }
}
```

The input field contains the following values:

```
Enter number of elements
5
Enter 5 integers
22
11
55
96
53
Enter value to find
22
```

The output field shows the result:

```
22 found at location 1.
```

The sidebar on the left includes links for "IDE", "My Projects", "Classroom", "Learn Programming", "Programming Questions", "Jobs", "Sign Up", and "Login". There is also a "GOT AN OPINION?" survey banner.

10) Write a C program to reverse a string?

Code:

```
#include <stdio.h>

#include <string.h>

int main()

{

    char s[100];

    printf("Enter a string to reverse\n");

    gets(s);

    strrev(s);

    printf("Reverse of the string: %s\n", s);

    return 0;

}
```

C:\Users\babue\OneDrive\Desktop\data structures\reverse a string.cpp - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit Release

(globals)

Project Class reverse a string.cpp

```
1 #include <stdio.h>
2
3 #include <string.h>
4
5 int main()
6 {
7     char s[100];
8
9     printf("Enter a string to reverse\n");
10
11    gets(s);
12
13    strrev(s);
14
15    printf("Reverse of the string: %s\n", s);
16
17    return 0;
18
19
20
21 }
```

C:\Users\babue\OneDrive\De

Enter a string to reverse
ambulance
Reverse of the string: ecnalubma

Process exited after 19.45 seconds with return value 0
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Errors: 0
Warnings: 0
Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\reverse a string.exe
Output Size: 323.1435546875 Kib
Compilation Time: 0.06s

Shorten compiler pat

Line: 21 Col: 21 Sel: 0 Lines: 21 Length: 225 Insert Done parsing in 0.141 seconds

11) Write a C program to check the string palindrome or not?

Code:

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

int main()
{
    char s[1000];
    int i,n,c=0;

    printf("Enter the string : ");
    gets(s);
    n=strlen(s);

    for(i=0;i<n/2;i++)
    {
        if(s[i]==s[n-i-1])
            c++;

    }

    if(c==i)
        printf("string is palindrome");
    else
        printf("string is not palindrome");

    return 0;
}
```

C Program To Check A String Is Palindrome | GDB online Debugger | Compiler | Online C Compiler

onlinegdb.com

Gmail YouTube Maps News Translate

Language: C

OnlineGDB beta
online compiler and debugger for c/c++

IDE My Projects Classroom Learn Programming Programming Questions Jobs Sign Up Login

GOT AN OPINION? SHARE AND GET REWARDED! OrkutunAP

Have fun taking surveys and get paid!

```
main.c
1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char s[100];
7     int i,n,c=0;
8
9     printf("Enter the string : ");
10    gets(s);
11    n=strlen(s);
12
13    for(i=0;i<n/2;i++)
14    {
15        if(s[i]==s[n-i-1])
16            c++;
17    }
18    if(c==i)
19        printf("string is palindrome");
20    else
21        printf("string is not palindrome");
22
23
24
25
26    return 0;
27 }
```

Input

Enter the string : tot
string is palindrome
...Program finished with exit code 0
Press ENTER to exit console.

12)Write a C program to count no.of vowels in a string ?

Code:

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    char str[100];
    int i, vowels = 0;
    printf("Enter the string: ");
    scanf("%[^\\n]s",&str);
    for(i = 0; str[i]; i++)
    {
        if(str[i]=='a' | | str[i]=='e' | | str[i]=='i' | |
           str[i]=='o' | | str[i]=='u' | | str[i]=='A' | |
           str[i]=='E' | | str[i]=='I' | | str[i]=='O' | | str[i]=='U')
        {
            vowels++;
        }
    }
    printf("Total number of vowels: = %d\\n",vowels);
    return 0;
}
```

The screenshot shows the OnlineGDB beta IDE interface. The main window displays a C program named 'main.c'.

```
main.c
1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     char str[100];
7     int i, vowels = 0;
8     printf("Enter the string: ");
9     scanf("%[^\n]s", &str);
10    for(i = 0; str[i]; i++)
11    {
12        if(str[i]=='a'|| str[i]=='e'|| str[i]=='i'|| str[i]=='o'|| str[i]=='u'|| str[i]=='A'|| str[i]=='E'|| str[i]=='I'|| str[i]=='O'|| str[i]=='U')
13        {
14            vowels++;
15        }
16    }
17    printf("Total number of vowels: = %d\n",vowels);
18
19 }
20
21
22
```

The code counts the total number of vowels in the input string 'babu'. The output window shows the result:

```
input
Enter the string: babu
Total number of vowels: = 4
...Program finished with exit code 0
Press ENTER to exit console.
```

13) Write a C program to find matrix Multiplication?

Code:

```
#include<stdio.h>

int main() {
    int a[10][10], b[10][10], c[10][10], n, i, j, k;
    printf("Enter the value of N (N <= 10): ");
    scanf("%d", &n);
    printf("Enter the elements of Matrix-A: \n");

    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }

    printf("Enter the elements of Matrix-B: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &b[i][j]);
        }
    }

    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            c[i][j] = 0;
            for (k = 0; k < n; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }

    printf("The product of the two matrices is: \n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
```

```

    printf("%d\t", c[i][j]);
}

printf("\n");

}

return 0;
}

```

```

main.c
1 #include<stdio.h>
2 int main() {
3     int a[10][10], b[10][10], c[10][10], n, i, j, k;
4     printf("Enter the value of N (N <= 10): ");
5     scanf("%d", &n);
6     printf("Enter the elements of Matrix-A: \n");
7
8     for (i = 0; i < n; i++) {
9         for (j = 0; j < n; j++) {
10            scanf("%d", &a[i][j]);
11        }
12    }
13    printf("Enter the elements of Matrix-B: \n");
14    for (i = 0; i < n; i++) {
15        for (j = 0; j < n; j++) {
16            scanf("%d", &b[i][j]);
17        }
18    }
19    for (i = 0; i < n; i++) {
20        for (j = 0; j < n; j++) {
21            c[i][j] = 0;
22            for (k = 0; k < n; k++) {
23                c[i][j] += a[i][k] * b[k][j];
24            }
25        }
26    }
27    printf("The product of the two matrices is: \n");
28    for (i = 0; i < n; i++) {
29        for (j = 0; j < n; j++) {
30            printf("%d\t", c[i][j]);
31        }
32        printf("\n");
33    }
34    return 0;
35 }

```

Enter the value of N (N <= 10): 3

Enter the elements of Matrix-A:

```

1
2
3
4
5
6
7
8
9

```

Enter the elements of Matrix-B:

```

1
2
3
4
5
6
7
8
9

```

The product of the two matrices is:

```

30   24   18
84   69   54
138  114   90

```

...Program finished with exit code 0
Press ENTER to exit console.

Day-3 data structures programs

1) Write a C program to implement linked list data structure?

Code:

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

struct Node {
    int data;
    struct Node* next;
};

void insert(struct Node** head, int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}

void deleteElement(struct Node** head, int value) {
    if (*head == NULL) {
        printf("List is empty.\n");
        return;
    }

    struct Node* current = *head;
    struct Node* previous = NULL;
    while (current != NULL && current->data != value) {
```

```
previous = current;
current = current->next;
}

if (current == NULL) {
    printf("Element not found in the list.\n");
    return;
}

if (previous == NULL) {
    *head = current->next;
} else {
    previous->next = current->next;
}
free(current);
printf("Element deleted successfully.\n");
}

void display(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}

int main() {
    struct Node* head = NULL;
    int choice, value;
    while (1) {
        printf("1. Insert an element\n");
        printf("2. Delete an element\n");
        printf("3. Display the list\n");
        printf("4. Exit\n");
    }
}
```

```
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
    case 1:
        printf("Enter the value to insert: ");
        scanf("%d", &value);
        insert(&head, value);
        break;
    case 2:
        printf("Enter the value to delete: ");
        scanf("%d", &value);
        deleteElement(&head, value);
        break;
    case 3:
        display(head);
        break;
    case 4:
        exit(0);
    default:
        printf("Invalid choice. Please try again.\n");}
return 0;
}
```

Result:

C:\Users\LENOVO L460\Downloads\kkjhgfds.cpp - [Executing] - Embarcadero Dev-C++ 6.3

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 9.2.0 64-bit

(globals)

Project C++ linked list.cpp link.cpp linked.cpp linked list implementation.cpp

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node* next;
6 };
7 void insert(struct Node** head, int value) {
8     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
9     newNode->data = value;
10    newNode->next = NULL;
11    if (*head == NULL) {
12        *head = newNode;
13    } else {
14        struct Node* current = *head;
15        while (current->next != NULL) {
16            current = current->next;
17        }
18        current->next = newNode;
19    }
20 }
21 void deleteElement(struct Node** head, int value) {
22     if (*head == NULL) {
23         printf("List is empty.\n");
24         return;
25     }
26     struct Node* current = *head;
27     struct Node* previous = NULL;
```

2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 10
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 29
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 30
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 1
Enter the value to insert: 40
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 2
Enter the value to delete: 29
Element deleted successfully.
1. Insert an element
2. Delete an element
3. Display the list
4. Exit
Enter your choice: 3
10 30 40
1. Insert an element
2. Delete an element
3. Display the list
4. Exit

Compiler (1) Resources Compile Log Debug Find Results Console Close

Abort Compilation

- Output Filename: C:\Users\LENOVO L460\Downloads\kkjhgfds.exe
- Output Size: 324.2373046875 Kib
- Compilation Time: 0.36s

Shorten compiler path

Line: 80 Col: 26 Sel: 0 Lines: 81 Length: 2257 Insert Done pasting

32°C Partly sunny 09:45 28-08-2023

2) Write a C program to merge two lists?

Code:

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

struct Node {
    int data;
    struct Node* next;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void insertEnd(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* temp = *head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

struct Node* mergeLists(struct Node* list1, struct Node* list2) {
    if (list1 == NULL) {
        return list2;
    }
    if (list2 == NULL) {
        return list1;
    }
```

```
    return list1;
}

if (list1->data < list2->data) {
    list1->next = mergeLists(list1->next, list2);
    return list1;
} else {
    list2->next = mergeLists(list1, list2->next);
    return list2;
}
}

void displayList(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct Node* list1 = NULL;
    struct Node* list2 = NULL;
    int n, m;
    printf("Enter the size of list 1: ");
    scanf("%d", &n);
    printf("Enter the elements of list 1:\n");
    for (int i = 0; i < n; i++) {
        int num;
        scanf("%d", &num);
        insertEnd(&list1, num);
    }
}
```

```

printf("Enter the size of list 1:");
scanf("%d", &m);
printf("Enter the elements of list 1:\n");
for (int i = 0; i < m; i++) {
    int num;
    scanf("%d", &num);
    insertEnd(&list1, num);
}
printf("List 1: ");
displayList(list1);
printf("List 2: ");
displayList(list2);
struct Node* mergedList = mergeLists(list1, list2);
printf("Merged List: ");
displayList(mergedList);
return 0;
}

```

The screenshot shows the Embarcadero Dev-C++ IDE interface. On the left, the code editor displays the source file `merge two lists.cpp` with the following content:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node* next;
6 };
7 struct Node* createNode(int data) {
8     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
9     newNode->data = data;
10    newNode->next = NULL;
11    return newNode;
12 }
13 void insertEnd(struct Node** head, int data) {
14     struct Node* newNode = createNode(data);
15     if (*head == NULL) {
16         *head = newNode;
17     } else {
18         struct Node* temp = *head;
19         while (temp->next != NULL) {
20             temp = temp->next;
21         }
22         temp->next = newNode;
23     }
24 }
25 struct Node* mergeLists(struct Node* list1, struct Node* list2) {
26     if (list1 == NULL) {
27         return list2;
28     }
29     if (list2 == NULL) {

```

The right side of the interface shows the terminal window with the following output:

```

C:\Users\LENOVO L460\Downloads\merge two lists - [Executing] - Embarcadero Dev-C++...
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit
(globals)
Project C > merge two lists.cpp
1 Enter the size of list 1: 5
2 Enter the elements of list 1:
3
4
5
6
7 Enter the size of list 2: 5
8 Enter the elements of list 2:
9
10
11
12
13 List 1: 1 -> 2 -> 3 -> 4 -> 5 -> NULL
14 List 2: 8 -> 7 -> 6 -> 9 -> 11 -> NULL
15 Merged List: 1 -> 2 -> 3 -> 4 -> 5 -> 8 -> 7 -> 6 -> 9 -> 11 -> NULL
16 -----
17 Process exited after 31.72 seconds with return value 0
18 Press any key to continue . . .

```

The status bar at the bottom shows the following information:

- Compiler Resources Compile Log Debug Find Results Console Close
- Output Filename: C:\Users\LENOVO L460\Downloads\merge two lists.exe
- Output Size: 324.1240234375 Kib
- Compilation Time: 1.75s
- 33°C Partly sunny 20:21 28-08-2023

3)Write a C program to implement stack operations?

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 4

int top = -1, inp_array[SIZE];

void push();
void pop();
void show();

int main()
{
    int choice;

    while (1)
    {
        printf("\nPerform operations on the stack:");

        printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");

        printf("\n\nEnter the choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                show();
                break;
            case 4:
                exit(0);
        }
    }
}
```

```
default:  
    printf("\nInvalid choice!!");  
}  
}  
}  
  
void push()  
{  
    int x;  
    if (top == SIZE - 1)  
    {  
        printf("\nOverflow!!");  
    }  
    else  
    {  
        printf("\nEnter the element to be added onto the stack: ");  
        scanf("%d", &x);  
        top = top + 1;  
        inp_array[top] = x;  
    }  
}  
  
void pop()  
{  
    if (top == -1)  
    {  
        printf("\nUnderflow!!");  
    }  
    else  
    {  
        printf("\nPopped element: %d", inp_array[top]);  
        top = top - 1;  
    }  
}
```

```
}
```

```
void show()
{
    if (top == -1)
    {
        printf("\nUnderflow!!");

    }
    else
    {
        printf("\nElements present in the stack: \n");
        for (int i = top; i >= 0; --i)
            printf("%d\n", inp_array[i]);
    }
}
```

The screenshot shows the Embarcadero Dev-C++ IDE interface. The project is named "stack op.cpp". The code editor displays the provided C++ code for a stack implementation. The compiler output window shows the execution of the program. The terminal window displays the following interaction:

```
Enter the element to be added onto the stack: 32
Perform operations on the stack:
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 1
Enter the element to be added onto the stack: 23
Perform operations on the stack:
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 2
Popped element: 23
Perform operations on the stack:
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 3
Elements present in the stack:
32
13
10
Perform operations on the stack:
1.Push the element
2.Pop the element
3.Show
4.End
```

The status bar at the bottom shows system information like temperature (33°C), weather (Partly sunny), date (28-08-2023), and time (20:41).

4)Write a C program to implement Queue data structure?

Code:

```
#include<stdio.h>
#define n 5
int main()
{
    int queue[n],ch=1,front=0,rear=0,i,j=1,x=n;
    printf("Queue using Array");
    printf("\n1.Insertion \n2.Deletion \n3.Display \n4.Exit");
    while(ch)
    {
        printf("\nEnter the Choice:");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                if(rear==x)
                    printf("\n Queue is Full");
                else
                {
                    printf("\nEnter no %d:",j++);
                    scanf("%d",&queue[rear++]);
                }
                break;
            case 2:
                if(front==rear)
                {
                    printf("\n Queue is empty");
                }
                else
                {
```

```
    printf("\n Deleted Element is %d",queue[front++]);  
    X++;  
}  
break;  
  
case 3:  
printf("\nQueue Elements are:\n ");  
if(front==rear)  
    printf("\n Queue is Empty");  
else  
{  
    for(i=front; i<rear; i++)  
    {  
        printf("%d",queue[i]);  
        printf("\n");  
    }  
    break;  
  
case 4:  
exit(0);  
  
default:  
    printf("Wrong Choice: please see the options");  
}  
}  
}  
return 0;  
}
```

Implementation of Queue using | merging two lists in c - Google | GDB online Debugger | Compile

onlinegdb.com

Gmail YouTube Maps News Translate

Run Debug Stop Share Save Beautify Language C

OnlineGDB beta
online compiler and debugger for c/c++

code, compile, run, debug, share.

IDE

My Projects Classroom new

Learn Programming Programming Questions

Jobs new Sign Up Login

GOT AN OPINION? SHARE AND SEE REWARDS! @RakutenAI

Have fun taking surveys and get paid!

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

main.c

```
16     {
17         printf("\n Enter no %d:",j++);
18         scanf("%d",&queue[rear++]);
19     }
20 }
```

input

Queue using Array

1.Insertion
2.Deletion
3.Display
4.Exit

Enter the Choice:

Enter no 1:10

Enter the Choice:

Enter no 2:20

Enter the Choice:

Enter no 3:30

Enter the Choice:

Enter no 4:40

Enter the Choice:

Deleted Element is 10

Enter the Choice:

Queue Elements are:

20
30
40

Enter the Choice:

5) Write a C program to convert infix to postfix Expression?

Code:

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

```
#include<ctype.h>
```

```
char stack[100];
```

```
int top = -1;
```

```
void push(char x)
```

```
{
```

```
    stack[++top] = x;
```

```
}
```

```
char pop()
```

```
{
```

```
    if(top == -1)
```

```
        return -1;
```

```
    else
```

```
        return stack[top--];
```

```
}
```

```
int priority(char x)
```

```
{
```

```
    if(x == '(')
```

```
        return 0;
```

```
    if(x == '+' || x == '-')
```

```
        return 1;
```

```
    if(x == '*' || x == '/')
```

```
        return 2;
```

```
    return 0;
```

```
}
```

```
int main()
{
    char exp[100];
    char *e, x;
    printf("Enter the expression : ");
    scanf("%s",exp);
    printf("\n");
    e = exp;

    while(*e != '\0')
    {
        if(isalnum(*e))
            printf("%c ",*e);
        else if(*e == '(')
            push(*e);
        else if(*e == ')')
        {
            while((x = pop()) != '(')
                printf("%c ",x);
        }
        else
        {
            while(priority(stack[top]) >= priority(*e))
                printf("%c ",pop());
            push(*e);
        }
        e++;
    }

    while(top != -1)
```

```
{  
    printf("%c ",pop());  
}  
return 0;  
}
```

The screenshot shows a web browser window for the OnlineGDB website (onlinedbg.com). The page displays a C program named 'main.c' for converting infix expressions to postfix. The code uses a stack represented by a character array 'exp'. It reads an expression from the user, processes it, and prints the resulting postfix expression. The browser's status bar indicates the date as 28-08-2023 and the time as 20:45. The system tray shows a weather icon for 30°C and 'Partly cloudy'.

```
main.c  
10     x || x == '-')  
11         return 1;  
12     if(x == '*' || x == '/')  
13         return 2;  
14     return 0;  
15 }  
16 int main()  
17 {  
18     char exp[100];  
19     char *e, x;  
20     printf("Enter the expression : ");  
21     scanf("%s",exp);  
22     printf("\n");  
23     e = exp;  
24     while(*e != '\0')  
25     {  
26         if(isalnum(*e))  
27             printf("%c",*e);  
28         else if(*e == '(')  
29             push(*e);  
30         else if(*e == ')')  
31             {  
32                 while((x = pop()) != '(')  
33                     printf("%c", x);  
34             }  
35         else  
36     }  
37 }  
38 ...Program finished with exit code 0  
39 Press ENTER to exit console.[]
```

Input:
Enter the expression : a+b*c
a b c * +
...Program finished with exit code 0
Press ENTER to exit console.[]

About • FAQ • Blog • Terms of Use • Contact Us • GDR

30°C
Partly cloudy

ENG IN 20:45 28-08-2023

6) write a C program to evaluate the postfix expression?

Code:

```
#include<stdio.h>

int stack[20];
int top = -1;
void push(int x)
{
    stack[++top] = x;
}

int pop()
{
    return stack[top--];
}

int main()
{
    char exp[20];
    char *e;
    int n1,n2,n3,num;
    printf("Enter the expression :: ");
    scanf("%s",exp);
    e = exp;
    while(*e != '\0')
    {
        if(isdigit(*e))
        {
            num = *e - 48;
            push(num);
        }
        else
```

```
{  
    n1 = pop();  
    n2 = pop();  
    switch(*e)  
    {  
        case '+':  
        {  
            n3 = n1 + n2;  
            break;  
        }  
        case '-':  
        {  
            n3 = n2 - n1;  
            break;  
        }  
        case '*':  
        {  
            n3 = n1 * n2;  
            break;  
        }  
        case '/':  
        {  
            n3 = n2 / n1;  
            break;  
        }  
    }  
    push(n3);  
}  
e++;  
}  
printf("\nThe result of expression %s = %d\n\n",exp,pop());
```

```
return 0;  
}
```

The screenshot shows the OnlineGDB IDE interface. The code editor contains the following C code:

```
main.c  
1 #include<stdio.h>  
2 int stack[20];  
3 int top = -1;  
4 void push(int x)  
5 {  
6     stack[++top] = x;  
7 }  
8  
9 int pop()  
10 {  
11     return stack[top--];  
12 }  
13  
14 int main()  
15 {  
16     char exp[20];  
17     char *e;  
18     int n1,n2,n3,num;  
19     printf("Enter the expression :: ");  
20     scanf("%s",exp);  
21     e = exp;  
22     while(*e != '\0')  
23     {  
24         if(isdigit(*e))  
25         {  
26             num = *e - 48;  
27             push(num);  
28         }  
29     else
```

The terminal window shows the output of the program:

```
input  
Enter the expression :: 245+*  
The result of expression 245+* = 18  
...Program finished with exit code 0  
Press ENTER to exit console.
```

7)Write a C program to implement tree traversals?

Code:

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

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

struct node* create(int x){
    struct node* newnode=(struct node*)malloc(sizeof(struct node));

    newnode->data=x;
    newnode->left=NULL;
    newnode->right=NULL;
}

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

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

```
}

void preorder(struct node* root){

    if(root==NULL){

        return;

    }

    printf("%d ",root->data);

    preorder(root->left);

    preorder(root->right);

}

int main(){

    int a;

    struct node* root = create(1);

    root->left=create(2);

    root->right=create(3);

    root->left->right=create(4);

    root->left->left=create(5);

    root->right->left=create(6);

    root->right->right=create(7);

    printf("enter the traversal type inorder->1 preorder->2 postorder->3:");

    scanf("%d",&a);

    switch(a){

        case 1:inorder(root);

        break;

        case 2:preorder(root);

        break;

        case 3:postorder(root);

        break;

    }

    return 0;

}
```

Binary Tree Traversal in Data Stru | Google | GDB online Debugger | Compile

onlinegdb.com

Translate

Language: C

IDE My Projects Classroom Learn Programming Programming Questions Jobs Sign Up Login

GOT AN OPINION? SHARE AND SEE REWARDS! @OkutaiAI

Have fun taking surveys and get paid!

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

```
main.c
1 #include <stdio.h>
2 #include<stdlib.h>
3 struct node{
4     int data;
5     struct node* left;
6     struct node* right;
7 };
8 struct node* create(int x){
9     struct node* newnode=(struct node*)malloc(sizeof(struct node));
10    newnode->data=x;
11    newnode->left=NULL;
12    newnode->right=NULL;
13 }
14 void inorder(struct node* root){
15     if(root==NULL){
16         return;
17     }
18     inorder(root->left);
19     printf("%d ",root->data);
20     inorder(root->right);
21 }
22 void postorder(struct node* root){
23     if(root==NULL){
24         return;
25     }
26 }
```

input

enter the traversal type inorder->1 preorder->2 postorder->3:1
5 2 4 1 6 3 7
...Program finished with exit code 0
Press ENTER to exit console.

Day 4: DATA STRUCTURES PROGRAMS

1.write a c program to implement binary search tree.

Program:

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

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

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}

struct Node* insert(struct Node* root, int data) {
    if (root == NULL) {
        return createNode(data);
    }
    if (data < root->data) {
        root->left = insert(root->left, data);
    } else if (data > root->data) {
        root->right = insert(root->right, data);
    }
    return root;
}

void inOrderTraversal(struct Node* root) {
    if (root != NULL) {
        inOrderTraversal(root->left);
```

```

printf("%d ", root->data);

inOrderTraversal(root->right);

}

}

int main() {

    struct Node* root = NULL;

    int n, value;

    printf("Enter the number of elements: ");

    scanf("%d", &n);

    printf("Enter the elements:\n");

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

        scanf("%d", &value);

        root = insert(root, value);

    }

    printf("In-order traversal of the binary search tree: ");

    inOrderTraversal(root);

    printf("\n");

    return 0;
}

```

The screenshot shows the Embarcadero Dev-C++ IDE interface during the execution of a program. The code in the editor is for an In-order traversal of a binary search tree. The terminal window displays the input and output. The input consists of 10 integers: 23, 32, 41, 22, 11, 56, 42, 49, 78, and 1. The output shows the In-order traversal of the tree, which prints the same 10 integers in ascending order: 1, 11, 22, 23, 32, 41, 42, 49, 56, 78.

```

C:\Users\LENOVO L460\Downloads\binary search tree.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2
(globals)
Project C binary search tree.cpp
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node* left;
6     struct Node* right;
7 };
8 struct Node* createNode(int data) {
9     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
10    newNode->data = data;
11    newNode->left = NULL;
12    newNode->right = NULL;
13    return newNode;
14 }
15 struct Node* insert(struct Node* root, int data) {
16     if (root == NULL) {
17         return createNode(data);
18     }
19     if (data < root->data) {
20         root->left = insert(root->left, data);
21     } else if (data > root->data) {
22         root->right = insert(root->right, data);
23     }
24     return root;
25 }
26 void inOrderTraversal(struct Node* root) {
27     if (root != NULL) {

```

Enter the number of elements: 10
Enter the elements:
23
32
11
56
42
49
78
In-order traversal of the binary search tree: 1 11 22 23 32 41 42 49 56 78

Process exited after 75.39 seconds with return value 0
Press any key to continue . . .

Compiler (1) Resources Compile Log Debug Find Results Console Close

Abort Compilation Output Filename: C:\Users\LENOVO L460\Downloads\binary search tree.exe
Output Size: 324.2587890625 Kib
Compilation Time: 0.72s

Shorten compiler path Line: 36 Col: 46 Sel: 0 Lines: 47 Length: 1250 Insert D

Type here to search 30°C Partly cloudy 09:21 29-08-2023

Sample input:elements:23,32,41,22,11,56,42,49,78,1 Output:inorder:1,11,22,23,32,41,42,49,56,78.

4.write a c program to implement bubble sort.

Program:

```
#include<stdio.h>

int main(){
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    int a[n];
    printf("Enter the elements: ");
    for(int i = 0; i < n; i++){
        scanf("%d", &a[i]);
    }
    int c=1;
    while(c<n){
        for(int j=0;j<n-c;j++){
            if(a[j]>a[j+1]){
                int temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
        c++;
    }
    for(int i=0;i<n;i++){
        printf("%d\t", a[i]);
    }
    return 0;
}
```

The screenshot shows a web-based C compiler interface from Programiz. The code editor contains a C program named 'main.c' that implements bubble sort. The output window shows the program's execution, including user input for array size and elements, and the sorted array. The interface includes tabs for other languages like Java, Python, and JavaScript, and a sidebar for C certification.

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     printf("Enter the size of the array: ");
5     scanf("%d", &n);
6     int a[n];
7     printf("Enter the elements: ");
8     for(int i = 0; i < n; i++){
9         scanf("%d", &a[i]);
10    }
11    int c=1;
12    while(c<n){
13        for(int j=0;j<n-c;j++){
14            if(a[j]>a[j+1]){
15                int temp=a[j];
16                a[j]=a[j+1];
17                a[j+1]=temp;
18            }
19        }
20    c++;
}
```

Processing request..

Output:

```
/tmp/KHNnhtwqQI.o
Enter the size of the array: 7
Enter the elements: 23 45 78 22 11 7 96
7 11 22 23 45 78 96 |
```

Sample input:

Enter the size of array: 7

Enter the elements: 23 45 78 22 11 7 96

Output:

7 11 22 23 45 78 96

5.write a c program to implement insertion sort.

Program:

```
#include <stdio.h>

int main(void)
{
    int n, i, j, temp;
    int arr[64];
    printf("Enter number of elements\n");
    scanf("%d", &n);
    printf("Enter %d integers\n", n);
    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    for (i = 1; i < n; i++)
    {
        j = i;
        while (j > 0 && arr[j - 1] > arr[j])
        {
            temp = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = temp;
            j--;
        }
    }
    printf("Sorted list in ascending order:\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\n", arr[i]);
    }
    return 0;
}
```

}

The screenshot shows the OnlineGDB IDE interface. On the left, there's a sidebar with links for IDE, My Projects, Classroom, Learn Programming, Programming Questions, Jobs, Sign Up, and Login. A promotional banner for "GOT AN OPINION? SHARE AND GET REWARDED! RakutenAP" is visible. The main area has tabs for "Binary Search Tree in C", "Online C Compiler", "GDB online Debugger | Compile", and "Insertion Sort Program in C". The "Insertion Sort Program in C" tab is active, displaying the following C code:

```
main.c
1 #include<stdio.h>
2 int main(){
3     int i, j, count, temp, number[25];
4     printf("Enter the size of the array: ");
5     scanf("%d",&count);
6     for(i=0;i<count;i++)
7         scanf("%d",&number[i]);
8     for(i=1;i<count;i++){
9         temp=number[i];
10        j=i-1;
11        while((temp>number[j])&&(j>=0)){
12            number[j+1]=number[j];
13            j=j-1;
14        }
15        number[j+1]=temp;
16    }
17    printf("Order of Sorted elements: ");
18    for(i=0;i<count;i++)
19        printf(" %d",number[i]);
20    return 0;
21 }
```

Below the code editor is a terminal window showing the execution of the program. The user enters "size:5", followed by "5 elements: 8 25 6 45 96", and the sorted output "Order of Sorted elements: 6 8 25 45 96". The terminal also displays "...Program finished with exit code 0" and "Press ENTER to exit console.[]".

Sample input: size:5

Integers: 34 21 73 1 22

Output: 1 21 22 34 73

6.write a c program to implement selection sort.

Program:

```
#include<stdio.h>

int main(){
    int i, j, count, temp, number[25];

    printf("How many numbers u are going to enter?: ");
    scanf("%d",&count);

    printf("Enter %d elements: ", count);
    for(i=0;i<count;i++)
        scanf("%d",&number[i]);

    for(i=0;i<count;i++){
        for(j=i+1;j<count;j++){
            if(number[i]>number[j]){
                temp=number[i];
                number[i]=number[j];
                number[j]=temp;
            }
        }
    }

    printf("Sorted elements: ");
    for(i=0;i<count;i++)
        printf(" %d",number[i]);
    return 0;
}
```

The screenshot shows the OnlineGDB IDE interface. The code editor contains a C program named 'main.c' for selection sort. The program prompts the user for the number of elements and then sorts them. The output window shows the sorted elements and the exit message.

```
1 #include<stdio.h>
2 int main(){
3     int i, j, count, temp, number[25];
4     print("How many numbers u are going to enter?: ");
5     scanf("%d",&count);
6     print("Enter %d elements: ", count);
7     for(i=0;i<count;i++)
8         scan("%d",&number[i]);
9     for(i=0;i<count;i++){
10        for(j=i+1;j<count;j++){
11            if(number[i]>number[j]){
12                temp=number[i];
13                number[i]=number[j];
14                number[j]=temp;
15            }
16        }
17    }
18    print("Sorted elements: ");
19    for(i=0;i<count;i++)
20        printf("%d",number[i]);
21    return 0;
22 }
```

Output:

```
How many numbers u are going to enter?: 5
Enter 5 elements: 45
46
23
85
280
Sorted elements: 23 45 46 85 280
...Program finished with exit code 0
Press ENTER to exit console.
```

Sample input: size 5

Integers: 18 4 23 9 19

Output: 4 9 18 19 23

7.write a c program to implement quick sort.

Program:

```
#include<stdio.h>

void quicksort(int number[25],int first,int last){
    int i, j, pivot, temp;

    if(first<last){
        pivot=first;
        i=first;
        j=last;

        while(i<j){
            while(number[i]<=number[pivot]&&i<last)
                i++;
            while(number[j]>number[pivot])
                j--;
            if(i<j){
                temp=number[i];
                number[i]=number[j];
                number[j]=temp;
            }
        }

        temp=number[pivot];
        number[pivot]=number[j];
        number[j]=temp;
        quicksort(number,first,j-1);
        quicksort(number,j+1,last);

    }
}
```

```

int main(){

    int i, count, number[25];

    printf("How many elements are u going to enter?: ");
    scanf("%d",&count);

    printf("Enter %d elements: ", count);
    for(i=0;i<count;i++)
        scanf("%d",&number[i]);


    quicksort(number,0,count-1);

    printf("Order of Sorted elements: ");
    for(i=0;i<count;i++)
        printf(" %d",number[i]);


    return 0;
}

```

The screenshot shows the OnlineGDB beta interface. The code in the editor is:

```

#include<stdio.h>
void quicksort(int number[25],int first,int last){
    int i, j, pivot, temp;
    if(first>last){
        pivot=first;
        i=first;
        j=last;
        while(i<j){
            while(number[i]<number[pivot]&&i<last)
                i++;
            while(number[j]>number[pivot])
                j--;
            if(i<j){
                temp=number[i];
                number[i]=number[j];
                number[j]=temp;
            }
        }
        temp=number[pivot];
        number[pivot]=number[j];
        number[j]=temp;
        quicksort(number,first,j-1);
        quicksort(number,j+1,last);
    }
}
int main(){
    int i, count, number[25];

```

The output window shows the following interaction:

```

Input:
How many elements are u going to enter?: 5
Enter 5 elements: 12
25
65
47
85

Output:
Order of Sorted elements: 12 25 47 65 85
...Program finished with exit code 0

```

Sample input: 12 25 65 47 85

Output: 12 25 47 65 85

8.write a c program to implement merge sort.

Program:

```
#include <stdio.h>

void merge(int arr[], int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];
    i = 0;
    j = 0;
    k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2) {
```

```
    arr[k] = R[j];
    j++;
    k++;
}
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    mergeSort(arr, 0, n - 1);
    printf("Sorted array:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}
```

The screenshot shows the Embarcadero Dev-C++ IDE interface. The main window displays a C++ code editor with a file named "merge sort.cpp". The code implements a merge sort algorithm. To the right of the editor is a terminal window showing the execution of the program. The user enters "10" as the number of elements, followed by ten integers: 25, 24, 58, 63, 85, 10, 43, 25, 633, and 52. The program outputs the sorted array: 10 24 25 25 43 52 58 63 85 633. Below the terminal window is a status bar showing compilation details: Errors: 0, Warnings: 0, Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\merge sort.exe, Output Size: 324.6994296875 KB, Compilation Time: 0.81s.

```
File Edit Search View Project Execute Tools ASyntax Window Help
C:\Users\babue\OneDrive\Desktop\data structures\merge sort.cpp - (Executing) - Embarcadero Dev-C++ 6.0
Project Class < > merge sort.cpp < Untitled1 >
File Edit Search View Project Execute Tools ASyntax Window Help
C:\Users\babue\OneDrive\Desktop\data structures\merge sort.cpp - (Executing) - Embarcadero Dev-C++ 6.0
TDM-GCC 9.2.0 64-bit Release
C:\Users\babue\OneDrive\Desktop\data structures\merge sort.cpp + v
C:\Users\babue\OneDrive\Desktop\data structures\merge sort.exe < + v
Enter the number of elements: 10
Enter 10 elements:
25
24
58
63
85
10
43
25
633
52
Sorted array:
10 24 25 25 43 52 58 63 85 633
Process exited after 30.23 seconds with return value 0
Press any key to continue . . .
Compiler Resources Help Compile Log < Debug C Find Results Console Close
Abort Compilation
Shorten compiler path
Errors: 0
Warnings: 0
Output Filename: C:\Users\babue\OneDrive\Desktop\data structures\merge sort.exe
Output Size: 324.6994296875 KB
Compilation Time: 0.81s
Line: 37 Col: 45 Sel: 0 Lines: 60 Length: 1351 Insert Done parsing in 0.032 seconds
```

Sample input:

Size=5

Elements= 23 11 46 22 9

Output: 9 11 22 23 46.

Day 5:

- 1) Write a c program to implement single source shortest algorithm(Dijkstra's)?

Code:

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX],int n,int startnode);
int main()
{
    int G[MAX][MAX],i,j,n,u;
    printf("Enter no. of vertices:");
    scanf("%d",&n);
    printf("\nEnter the adjacency matrix:\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&G[i][j]);
    printf("\nEnter the starting node:");
    scanf("%d",&u);
    dijkstra(G,n,u);
    return 0;
}
void dijkstra(int G[MAX][MAX],int n,int startnode)
{
    int cost[MAX][MAX],distance[MAX],pred[MAX];
    int visited[MAX],count,mindistance,nextnode,i,j;

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            if(G[i][j]==0)
                cost[i][j]=INFINITY;
            else
                cost[i][j]=G[i][j];

    for(i=0;i<n;i++)
    {
        distance[i]=cost[startnode][i];
        pred[i]=startnode;
        visited[i]=0;
    }
    distance[startnode]=0;
    visited[startnode]=1;
    count=1;
    while(count<n-1)
    {
        mindistance=INFINITY;
```

```

for(i=0;i<n;i++)
if(distance[i]<mindistance&&!visited[i])
{
mindistance=distance[i];
nextnode=i;
}

visited[nextnode]=1;
for(i=0;i<n;i++)
if(!visited[i])
if(mindistance+cost[nextnode][i]<distance[i])
{
distance[i]=mindistance+cost[nextnode][i];
pred[i]=nextnode;
}
count++;
}

for(l=0;l<n;l++)
if(l!=startnode)
{
printf("\nDistance of node%d=%d",l,distance[l]);
printf("\nPath=%d",l);
j=l;
do
{
j=pred[j];
printf("<-%d",j);
}while(j!=startnode);
}
}

```


- 2) Write a c program to implement minimum spanning tree from prim's algorithm?

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#define V 5
int minKey(int key[], int mstSet[]) {
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++) {
        if (!mstSet[v] && key[v] < min) {
            min = key[v];
            min_index = v;
        }
    }
    return min_index;
}
void printMST(int parent[], int graph[V][V]) {
    printf("Edge  Weight\n");
    for (int i = 1; i < V; i++) {
        printf("%d - %d  %d \n", parent[i], i, graph[i][parent[i]]);
    }
}
void primMST(int graph[V][V]) {
    int parent[V];
    int key[V];
    int mstSet[V];
    for (int i = 0; i < V; i++) {
        key[i] = INT_MAX;
        mstSet[i] = 0;
    }
    key[0] = 0;
    parent[0] = -1;
    for (int count = 0; count < V - 1; count++) {
        int u = minKey(key, mstSet);
        mstSet[u] = 1;
        for (int v = 0; v < V; v++) {
            if (graph[u][v] && !mstSet[v] && graph[u][v] < key[v]) {
                parent[v] = u;
                key[v] = graph[u][v];
            }
        }
    }
    printMST(parent, graph);
}
int main() {
    int graph[V][V];
    printf("Enter the adjacency matrix for the graph:\n");
    for (int i = 0; i < V; i++) {
```

```

        for (int j = 0; j < V; j++) {
            scanf("%d", &graph[i][j]);
        }
    }
    primMST(graph);
    return 0;
}

```

The screenshot shows the OnlineGDB beta interface. On the left, there's a sidebar with links for IDE, My Projects, Classroom, Learn Programming, Programming Questions, Jobs, Sign Up, Login, and Learn Python with KodeKloud. The main area has tabs for "Prims Algorithm" and "Prim's MST Algorithm in C". The code editor contains a file named "main.c" with the provided C code for Prim's algorithm. Below the code editor is a terminal window titled "Input" where an adjacency matrix is entered:

```

Enter the adjacency matrix for the graph:
0 10 0 30 100
10 0 50 0 0
0 50 0 20 10
30 0 20 0 60
100 0 0 60 0
Edge   Weight
0 - 1   10
3 - 2   20
0 - 3   30
2 - 4   10

```

At the bottom of the terminal, it says "...Program finished with exit code 0 Press ENTER to exit console. []".

- 3) Write a c program to implement Kruskal's algorithm?

Code:

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

int comparator(const void* p1, const void* p2)
{
    const int(*x)[3] = p1;
    const int(*y)[3] = p2;
    return (*x)[2] - (*y)[2];
}

void makeSet(int parent[], int rank[], int n)
{
    for (int i = 0; i < n; i++) {
        parent[i] = i;
        rank[i] = 0;
    }
}

int findParent(int parent[], int component)
{
    if (parent[component] == component)
        return component;
    return parent[component]
        = findParent(parent, parent[component]);
}

void unionSet(int u, int v, int parent[], int rank[], int n)
{
    u = findParent(parent, u);
    v = findParent(parent, v);
    if (rank[u] < rank[v]) {
        parent[u] = v;
    }
    else if (rank[u] > rank[v]) {
        parent[v] = u;
    }
    else {
        parent[v] = u;
        rank[u]++;
    }
}

void kruskalAlgo(int n, int edge[n][3])
{
```

```

qsort(edge, n, sizeof(edge[0]), comparator);
int parent[n];
int rank[n];

makeSet(parent, rank, n);

int minCost = 0;
printf(
"Following are the edges in the constructed MST\n");
for (int i = 0; i < n; i++) {
int v1 = findParent(parent, edge[i][0]);
int v2 = findParent(parent, edge[i][1]);
int wt = edge[i][2];

if (v1 != v2) {
unionSet(v1, v2, parent, rank, n);
minCost += wt;
printf("%d -- %d == %d\n", edge[i][0],
edge[i][1], wt);
}
}
printf("Minimum Cost Spanning Tree: %d\n", minCost);
}

int main()
{
int edge[5][3] = { { 0, 1, 10 },
{ 0, 2, 6 },
{ 0, 3, 5 },
{ 1, 3, 15 },
{ 2, 3, 4 } };
kruskalAlgo(5, edge);
return 0;
}

```

The screenshot shows a web-based IDE interface for OnlineGDB beta. The main area displays a C program named 'main.c' with code for implementing Prim's Minimum Spanning Tree algorithm. The code includes functions for comparator logic, initializing sets, finding parents, and performing union operations. The output window shows the edges in the constructed MST and the minimum cost of 19. The interface includes a toolbar with 'Run', 'Debug', 'Stop', 'Save', and 'Beautify' buttons, and a sidebar with various programming-related links.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int comparator(const void* p1, const void* p2)
5 {
6     const int(*x)[2] = p1;
7     const int(*y)[2] = p2;
8     return (*x)[1] - (*y)[1];
9 }
10
11 void makeSet(int parent[], int rank[], int n)
12 {
13     for (int i = 0; i < n; i++) {
14         parent[i] = i;
15         rank[i] = 0;
16     }
17 }
18
19 int findParent(int parent[], int component)
20 {
21     if (parent[component] == component)
22         return component;
23     return parent[component];
24     = findParent(parent, parent[component]);
25 }
26
27 void unionSet(int u, int v, int parent[], int rank[], int n)
28 {
29     if (rank[u] > rank[v])
30         parent[v] = u;
31     else if (rank[u] < rank[v])
32         parent[u] = v;
33     else {
34         parent[v] = u;
35         rank[u]++;
36     }
37 }
38
39 void printMST(int parent[], int n)
40 {
41     printf("Following are the edges in the constructed MST\n");
42     for (int i = 1; i < n; i++) {
43         if (parent[i] != i) {
44             printf("%d -- %d == %d\n", i, parent[i], i);
45         }
46     }
47 }
48
49 int main()
50 {
51     int V = 5;
52     int E = 8;
53     int parent[V];
54     int rank[V];
55     int graph[V][V] = {
56         {0, 1, 2}, {0, 2, 3}, {0, 3, 4}, {1, 2, 1}, {1, 3, 2}, {1, 4, 3}, {2, 3, 1}, {2, 4, 2}, {3, 4, 1}
57     };
58
59     // A utility function to print MST stored in parent[] and rank[]
60     printMST(parent, V);
61
62     // Call the MST() function
63     MST(graph, V, parent, rank);
64
65     // Print the constructed MST
66     printMST(parent, V);
67
68     return 0;
69 }
```

Following are the edges in the constructed MST
2 -- 3 == 4
0 -- 3 == 5
0 -- 1 == 10
Minimum Cost Spanning Tree: 19

...Program finished with exit code 0
Press ENTER to exit console.

- 4) Write a c program to implement the graph traversals?

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int vertex;
    struct node *next;
};
struct node *createNode(int);
struct Graph
{
    int numVertices;
    struct node **adjLists;
    int *visited;
};
struct Graph *createGraph(int vertices)
{
    struct Graph *graph = malloc(sizeof(struct Graph));
    graph->numVertices = vertices;
    graph->adjLists = malloc(vertices * sizeof(struct node *));
    graph->visited = malloc(vertices * sizeof(int));
    int i;
    for (i = 0; i < vertices; i++)
    {
        graph->adjLists[i] = NULL;
        graph->visited[i] = 0;
    }
    return graph;
}
void addEdge(struct Graph *graph, int src, int dest)
{
    struct node *newNode = createNode(dest);
    newNode->next = graph->adjLists[src];
    graph->adjLists[src] = newNode;
    newNode = createNode(src);
    newNode->next = graph->adjLists[dest];
    graph->adjLists[dest] = newNode;
}
struct node *createNode(int v)
{
    struct node *newNode = malloc(sizeof(struct node));
    newNode->vertex = v;
    newNode->next = NULL;
    return newNode;
}
void printGraph(struct Graph *graph)
{
```

```

int v;
for (v = 0; v < graph->numVertices; v++)
{
    struct node *temp = graph->adjLists[v];
    printf("\n Adjacency list of vertex %d\n ", v);
    while (temp)
    {
        printf("%d -> ", temp->vertex);
        temp = temp->next;
    }
    printf("\n");
}
}

void bfs(struct Graph *graph, int startVertex)
{
    struct node *queue = NULL;
    graph->visited[startVertex] = 1;
    enqueue(&queue, startVertex);
    while (!isEmpty(queue))
    {
        printQueue(queue);
        int currentVertex = dequeue(&queue);
        printf("Visited %d ", currentVertex);
        struct node *temp = graph->adjLists[currentVertex];
        while (temp)
        {
            int adjVertex = temp->vertex;
            if (graph->visited[adjVertex] == 0)
            {
                graph->visited[adjVertex] = 1;
                enqueue(&queue, adjVertex);
            }
            temp = temp->next;
        }
    }
}

int isEmpty(struct node *queue)
{
    return queue == NULL;
}

void enqueue(struct node **queue, int value)
{
    struct node *newNode = createNode(value);
    if (isEmpty(*queue))
    {
        *queue = newNode;
    }
    else

```

```

{
struct node *temp = *queue;
while (temp->next)
{
temp = temp->next;
}
temp->next = newNode;
}
}

int dequeue(struct node **queue)
{
int nodeData = (*queue)->vertex;
struct node *temp = *queue;
*queue = (*queue)->next;
free(temp);
return nodeData;
}

void printQueue(struct node *queue)
{
while (queue)
{
printf("%d ", queue->vertex);
queue = queue->next;
}
printf("\n");
}

int main(void)
{
struct Graph *graph = createGraph(6);
printf("\nWhat do you want to do?\n");
printf("1. Add edge\n");
printf("2. Print graph\n");
printf("3. BFS\n");
printf("4. Exit\n");
int choice;
scanf("%d", &choice);
while (choice != 4)
{
if (choice == 1)
{
int src, dest;
printf("Enter source and destination: ");
scanf("%d %d", &src, &dest);
addEdge(graph, src, dest);
}
else if (choice == 2)
{
printGraph(graph);
}
}
}

```

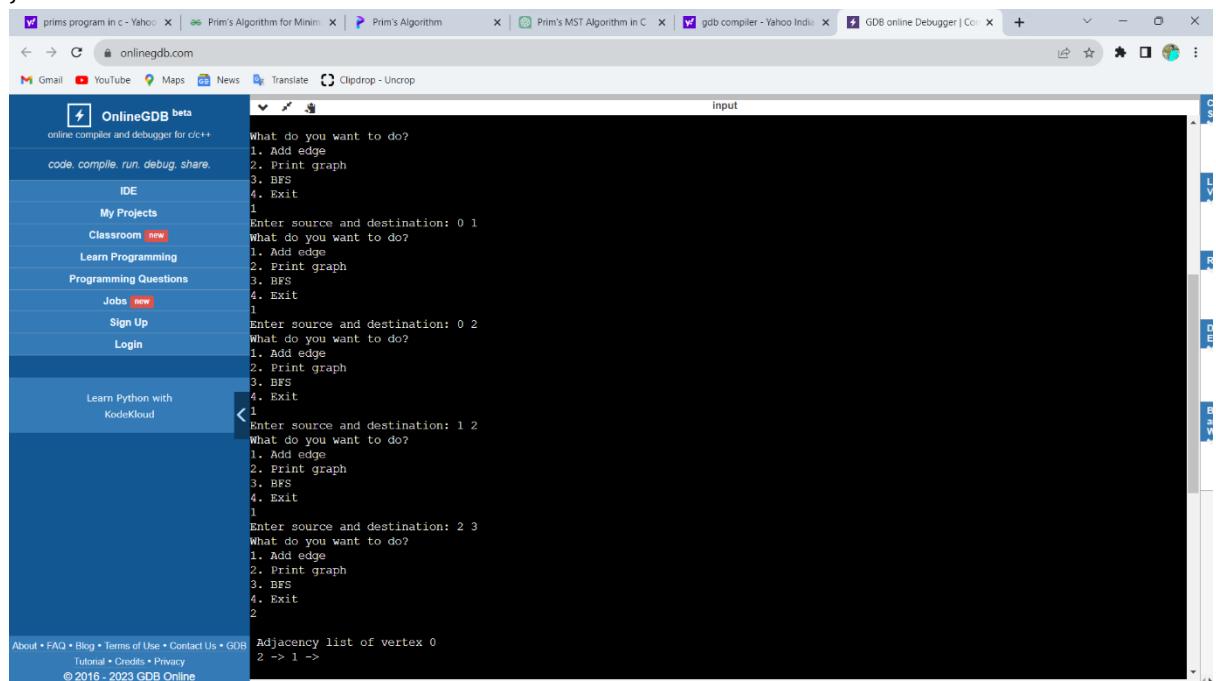
```

}

else if (choice == 3)
{
    int startVertex;
    printf("Enter starting vertex: ");
    scanf("%d", &startVertex);
    bfs(graph, startVertex);
}
else
{
    printf("Invalid choice\n");
}
printf("What do you want to do?\n");
printf("1. Add edge\n");
printf("2. Print graph\n");
printf("3. BFS\n");
printf("4. Exit\n");
scanf("%d", &choice);
}

return 0;
}

```



The screenshot shows a browser window with multiple tabs open, including "prims program in c - Yahoo!", "Prim's Algorithm for Minim... - Yahoo!", "Prim's Algorithm - Yahoo!", "Prim's MST Algorithm in C - Yahoo!", "gdb compiler - Yahoo! India", and "GDB online Debugger | Co... - Yahoo!". The main content area displays the OnlineGDB IDE interface.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Jobs new

Sign Up

Login

Learn Python with KodeKloud

OnlineGDB beta

online compiler and debugger for c/c++

code, compile, run, debug, share.

input

```
2. Print graph
3. BFS
4. Exit
2

Adjacency list of vertex 0
2 -> 1 ->

Adjacency list of vertex 1
2 -> 0 ->

Adjacency list of vertex 2
3 -> 1 -> 0 ->

Adjacency list of vertex 3
2 ->

Adjacency list of vertex 4

Adjacency list of vertex 5

What do you want to do?
1. Add edge
2. Print graph
3. BFS
4. Exit
3
Enter starting vertex: 0
0
Visited 0 2 1
Visited 2 1 3
Visited 1 3
Visited 3 What do you want to do?
1. Add edge
2. Print graph
3. BFS
4. Exit
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2023 GDB Online

