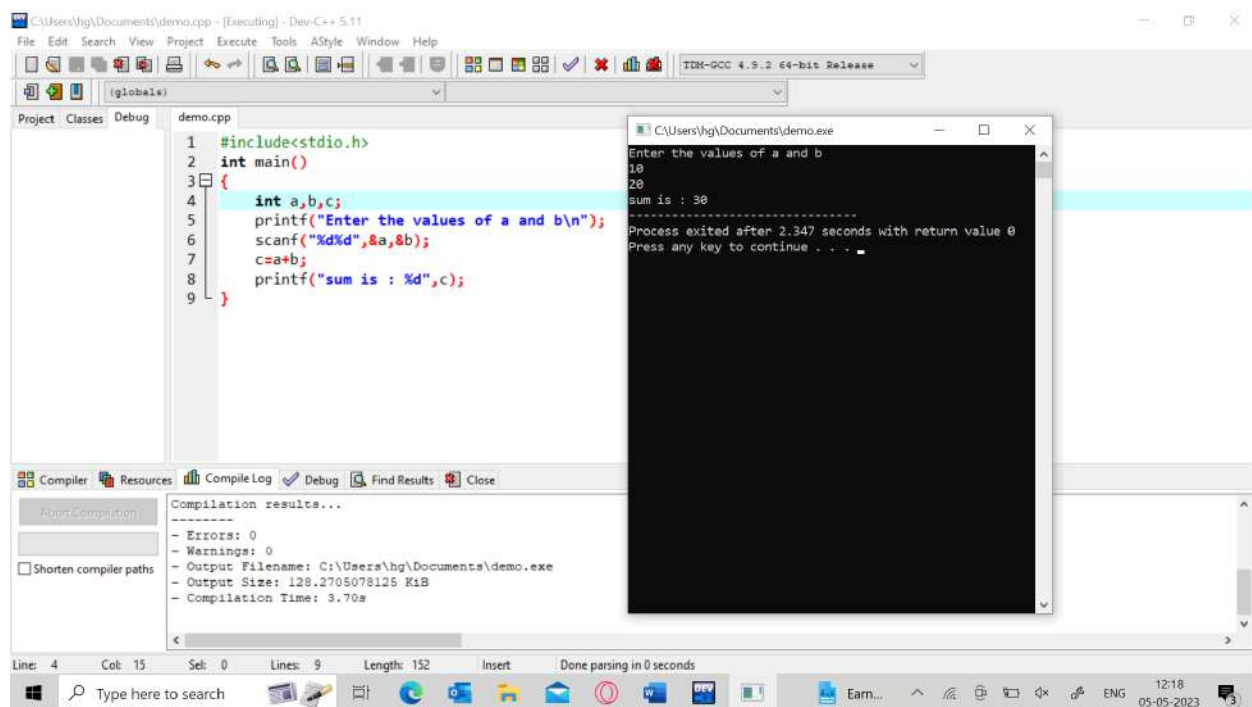


Name : J. Joel Andrew

Register Number : 192211111

CSA0323 DATA STRUCTURES WITH MULTITASK PROGRAMMING

1) ADDITION IN C



The screenshot displays a C++ IDE with a project named 'demo.cpp'. The code in the editor is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     printf("Enter the values of a and b\n");
6     scanf("%d%d",&a,&b);
7     c=a+b;
8     printf("sum is : %d",c);
9 }
```

The program is compiled and executed. The output window shows the following text:

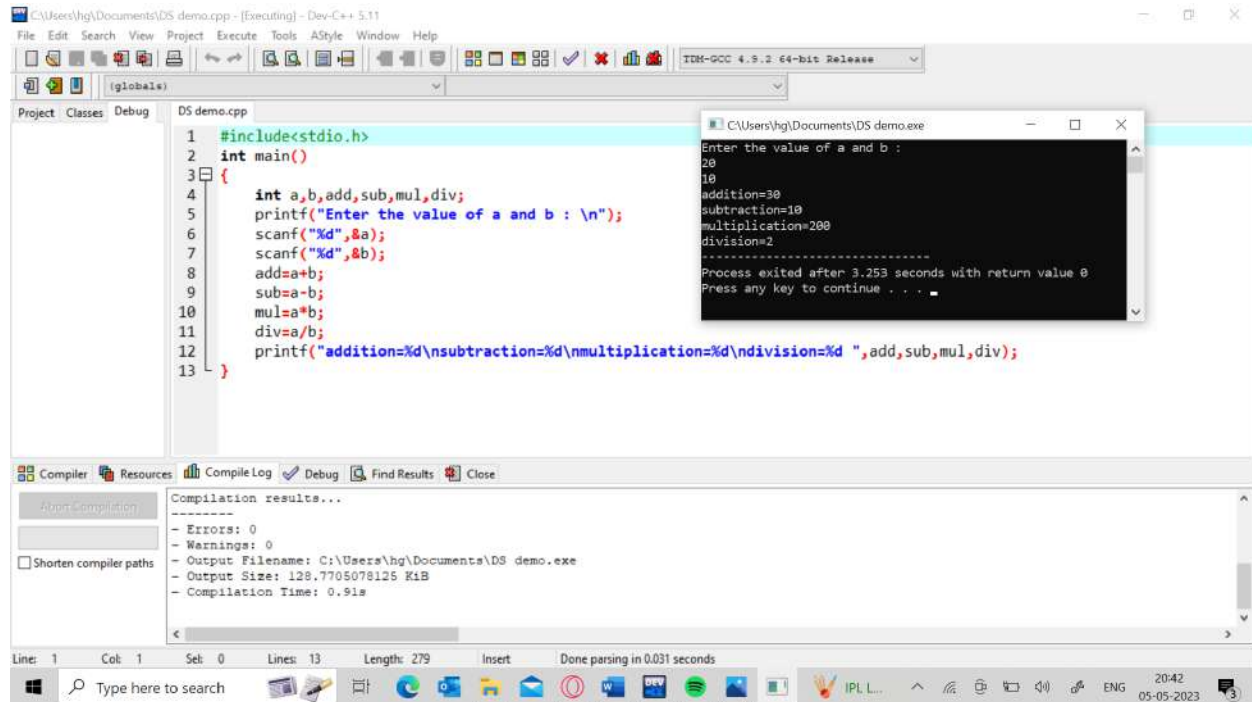
```
Enter the values of a and b
10
20
sum is : 30
.....
Process exited after 2.347 seconds with return value 0
Press any key to continue . . .
```

The IDE also shows the compilation results in the bottom panel:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\demo.exe
- Output Size: 128,270,507,8125 KiB
- Compilation Time: 3.70s
```

The status bar at the bottom indicates the current line is 4, column is 15, and the file length is 152 characters.

2) ARITHMETIC OPERATIONS IN C USING DYNAMIC INITIALIZATION



The screenshot shows the Dev-C++ IDE with a C program that uses dynamic initialization. The code in `DS demo.cpp` is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,add,sub,mul,div;
5     printf("Enter the value of a and b : \n");
6     scanf("%d",&a);
7     scanf("%d",&b);
8     add=a+b;
9     sub=a-b;
10    mul=a*b;
11    div=a/b;
12    printf("addition=%d\nsubtraction=%d\nmultiplication=%d\ndivision=%d ",add,sub,mul,div);
13 }
```

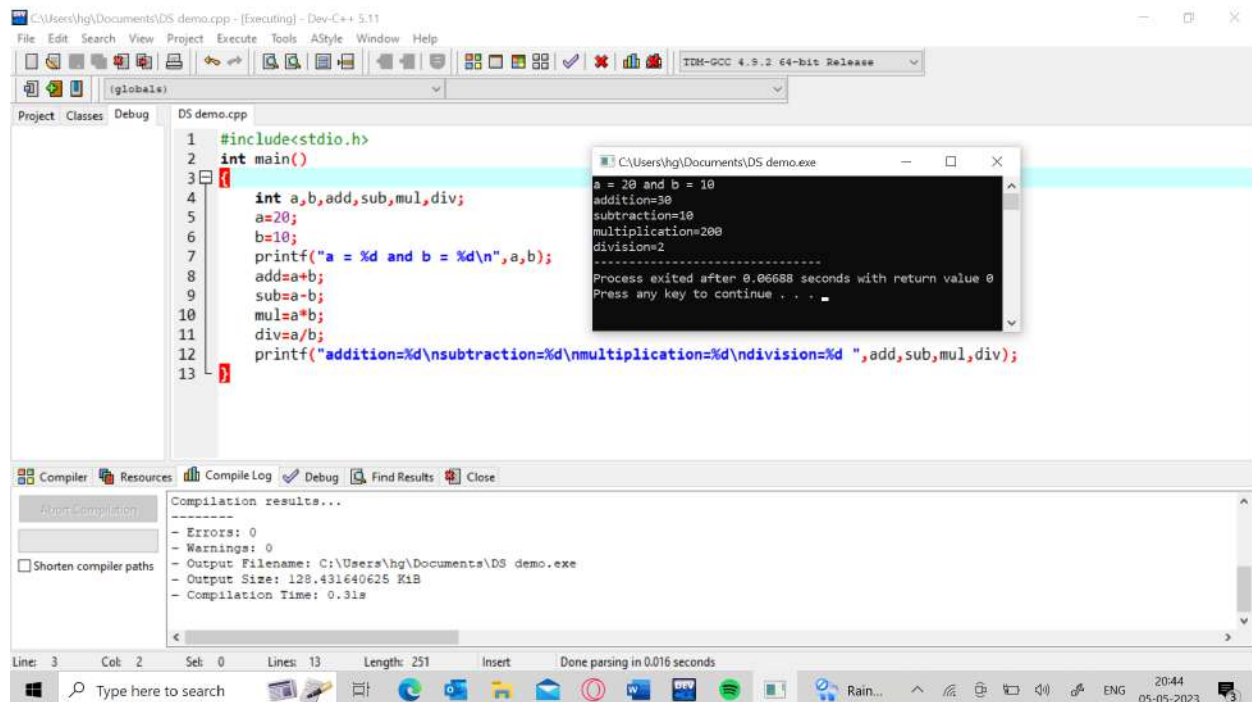
The output window `C:\Users\hg\Documents\DS demo.exe` shows the following output:

```
Enter the value of a and b :
20
10
addition=30
subtraction=10
multiplication=200
division=2
.....
Process exited after 3.253 seconds with return value 0
Press any key to continue . . .
```

The compilation results window shows:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS demo.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.91s
```

3) ARITHMETIC OPERATION IN C USING STATIC INITIALIZATION



The screenshot shows the Dev-C++ IDE with a C program that uses static initialization of variables `a` and `b`. The code in `DS demo.cpp` is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,add,sub,mul,div;
5     a=20;
6     b=10;
7     printf("a = %d and b = %d\n",a,b);
8     add=a+b;
9     sub=a-b;
10    mul=a*b;
11    div=a/b;
12    printf("addition=%d\nsubtraction=%d\nmultiplication=%d\ndivision=%d ",add,sub,mul,div);
13 }
```

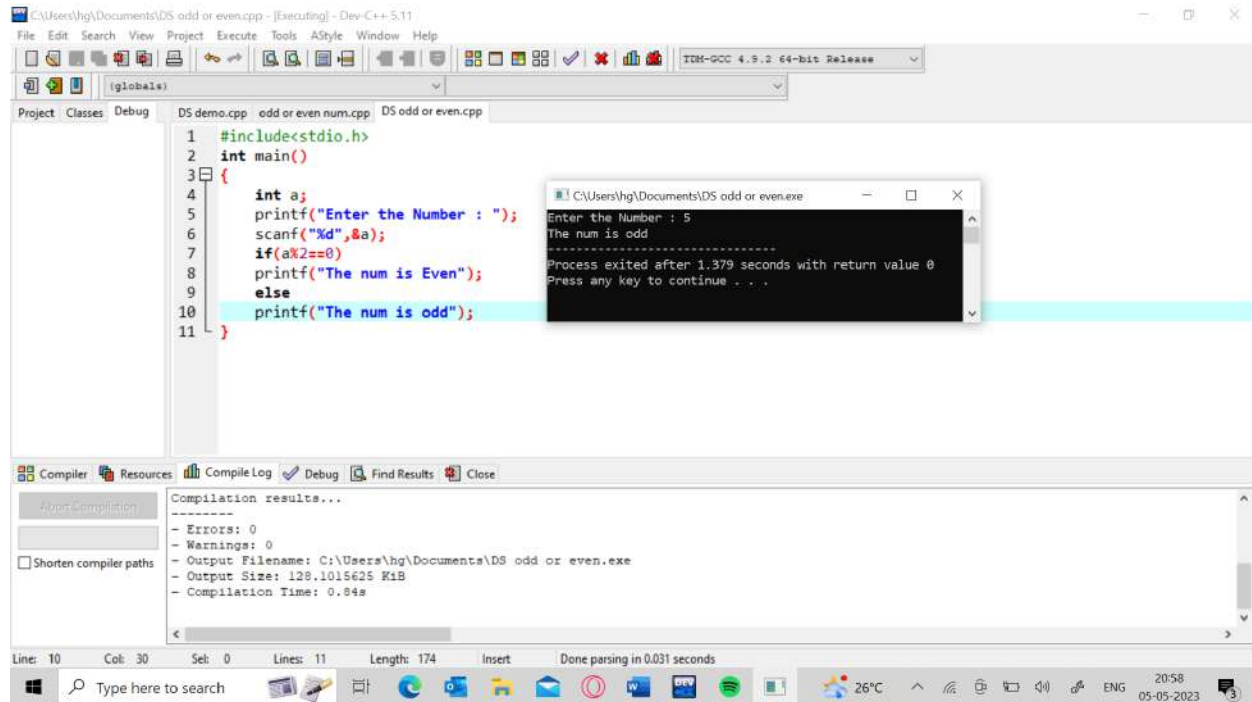
The output window `C:\Users\hg\Documents\DS demo.exe` shows the following output:

```
a = 20 and b = 10
addition=30
subtraction=10
multiplication=200
division=2
.....
Process exited after 0.06688 seconds with return value 0
Press any key to continue . . .
```

The compilation results window shows:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS demo.exe
- Output Size: 128.431640625 KiB
- Compilation Time: 0.31s
```

4) C PROGRAM TO CHECK THE GIVEN NUMBER IS ODD OR EVEN



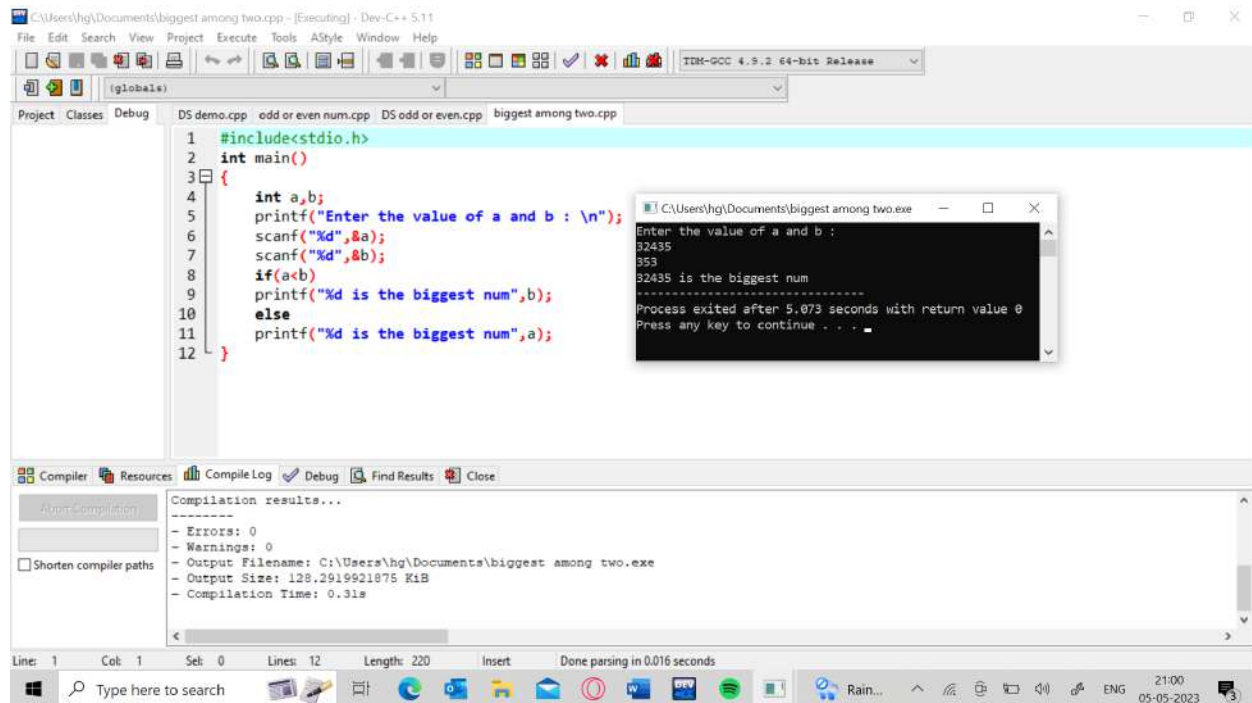
```
1 #include<stdio.h>
2 int main()
3 {
4     int a;
5     printf("Enter the Number : ");
6     scanf("%d",&a);
7     if(a%2==0)
8         printf("The num is Even");
9     else
10        printf("The num is odd");
11 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS odd or even.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.84s

Enter the Number : 5
The num is odd
.....
Process exited after 1.379 seconds with return value 0
Press any key to continue . . .

5) CHECKING THE BIGGEST BETWEEN TWO NUMBERS



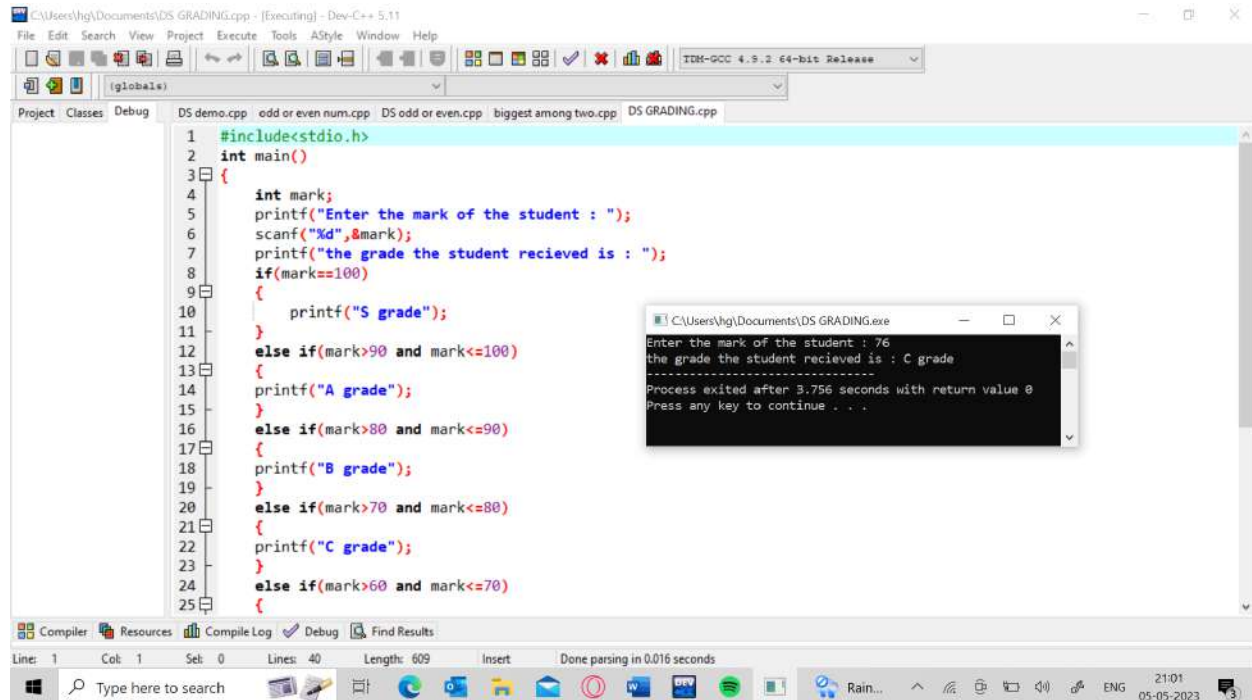
```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b;
5     printf("Enter the value of a and b : \n");
6     scanf("%d",&a);
7     scanf("%d",&b);
8     if(a<b)
9         printf("%d is the biggest num",b);
10    else
11        printf("%d is the biggest num",a);
12 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\biggest among two.exe
- Output Size: 128.2919921875 KiB
- Compilation Time: 0.31s

Enter the value of a and b :
32435
353
32435 is the biggest num
.....
Process exited after 5.073 seconds with return value 0
Press any key to continue . . .

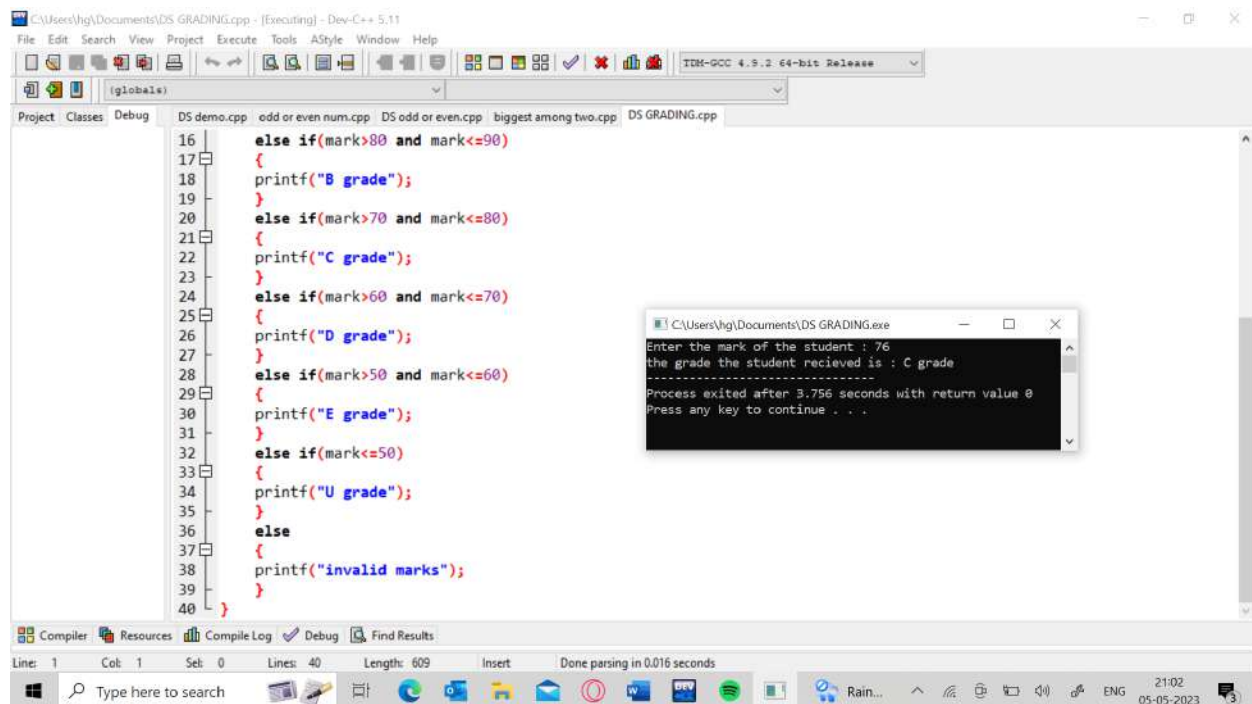
6) MARKS AND GRADES IN C (USING CONDITIONAL STATEMENTS)



```
1 #include<stdio.h>
2 int main()
3 {
4     int mark;
5     printf("Enter the mark of the student : ");
6     scanf("%d",&mark);
7     printf("the grade the student recieved is : ");
8     if(mark==100)
9     {
10        printf("S grade");
11    }
12    else if(mark>90 and mark<=100)
13    {
14        printf("A grade");
15    }
16    else if(mark>80 and mark<=90)
17    {
18        printf("B grade");
19    }
20    else if(mark>70 and mark<=80)
21    {
22        printf("C grade");
23    }
24    else if(mark>60 and mark<=70)
25    {
```

Console Output:

```
C:\Users\hgh\Documents\DS GRADING.exe
Enter the mark of the student : 76
the grade the student recieved is : C grade
.....
Process exited after 3.756 seconds with return value 0
Press any key to continue . . .
```

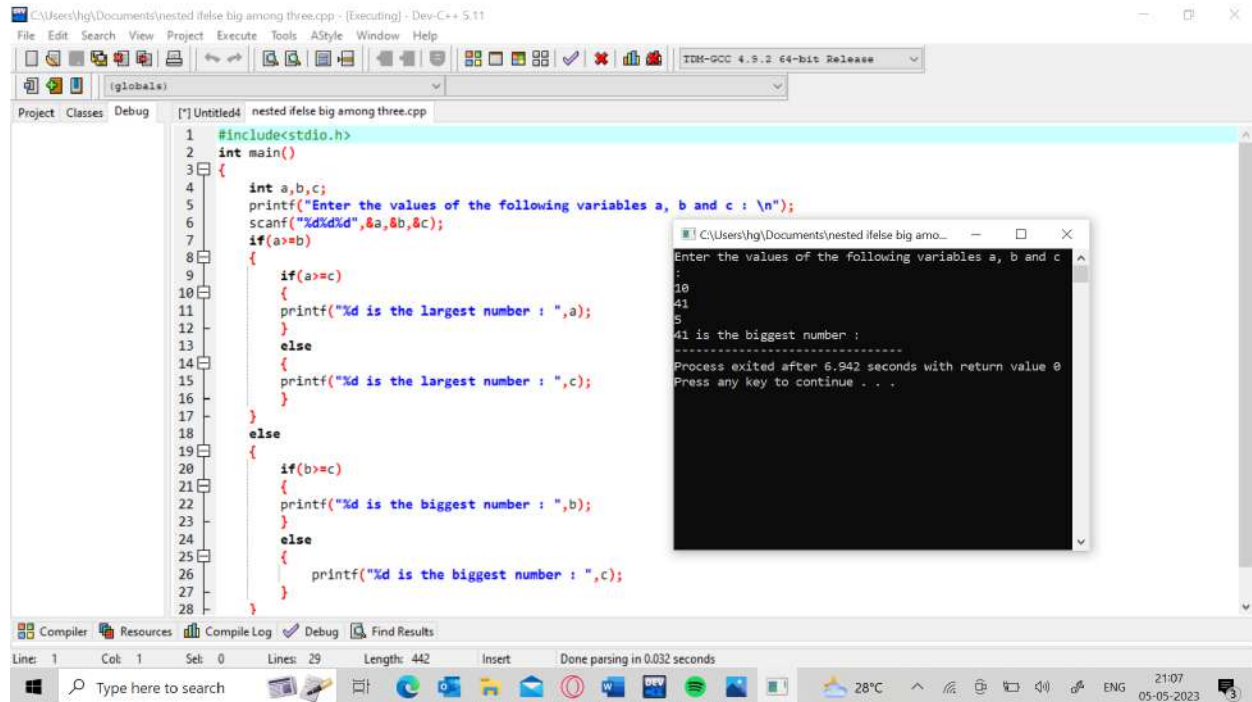


```
16    else if(mark>80 and mark<=90)
17    {
18        printf("B grade");
19    }
20    else if(mark>70 and mark<=80)
21    {
22        printf("C grade");
23    }
24    else if(mark>60 and mark<=70)
25    {
26        printf("D grade");
27    }
28    else if(mark>50 and mark<=60)
29    {
30        printf("E grade");
31    }
32    else if(mark<=50)
33    {
34        printf("U grade");
35    }
36    else
37    {
38        printf("invalid marks");
39    }
40 }
```

Console Output:

```
C:\Users\hgh\Documents\DS GRADING.exe
Enter the mark of the student : 76
the grade the student recieved is : C grade
.....
Process exited after 3.756 seconds with return value 0
Press any key to continue . . .
```

7) BIGGER BETWEEN THREE VALUES



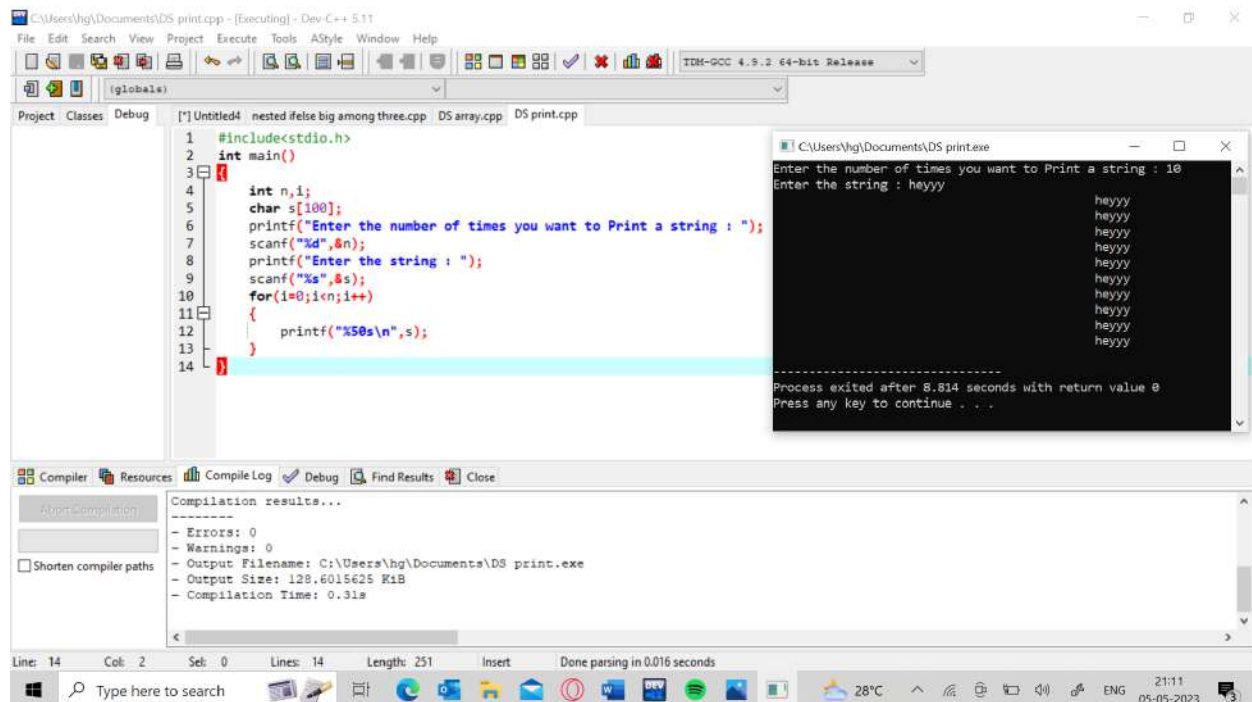
The screenshot shows the Dev-C++ IDE with a C++ program open. The program prompts the user to enter three values (a, b, and c) and then determines the largest number using a series of if-else statements. The execution window shows the user inputting 10, 41, and 5, with the program correctly identifying 41 as the largest number.

```
1 #include<stdio.h>
2 int main()
3 {
4     int a,b,c;
5     printf("Enter the values of the following variables a, b and c : \n");
6     scanf("%d%d%d",&a,&b,&c);
7     if(a>=b)
8     {
9         if(a>=c)
10        {
11            printf("%d is the largest number : ",a);
12        }
13        else
14        {
15            printf("%d is the largest number : ",c);
16        }
17    }
18    else
19    {
20        if(b>=c)
21        {
22            printf("%d is the biggest number : ",b);
23        }
24        else
25        {
26            printf("%d is the biggest number : ",c);
27        }
28    }
```

Execution Output:

```
Enter the values of the following variables a, b and c :
10
41
5
41 is the biggest number :
-----
Process exited after 6.942 seconds with return value 0
Press any key to continue . . .
```

8) PRINT STATEMENT USING LOOPING



The screenshot shows the Dev-C++ IDE with a C++ program open. The program prompts the user to enter the number of times to print a string and the string itself. It then uses a for loop to print the string the specified number of times. The execution window shows the user inputting 10 and 'heyyy', with the program printing 'heyyy' 10 times.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,i;
5     char s[100];
6     printf("Enter the number of times you want to Print a string : ");
7     scanf("%d",&n);
8     printf("Enter the string : ");
9     scanf("%s",&s);
10    for(i=0;i<n;i++)
11    {
12        printf("%50s\n",s);
13    }
14 }
```

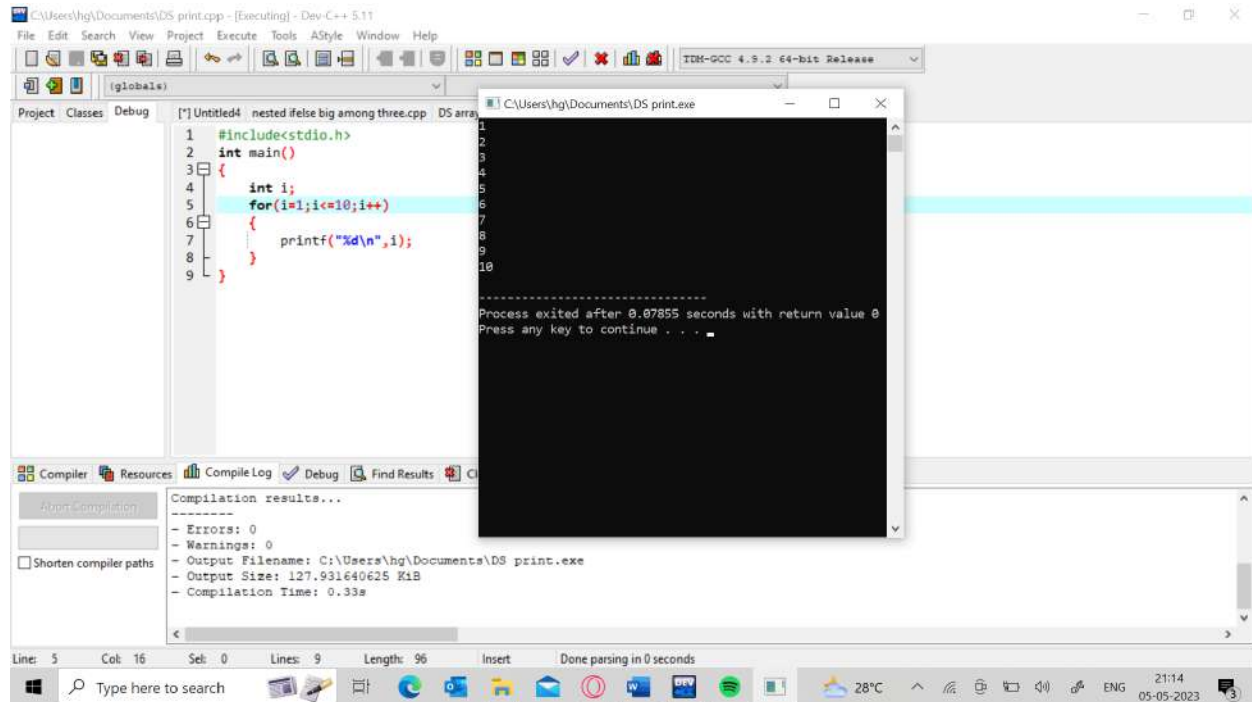
Execution Output:

```
Enter the number of times you want to Print a string : 10
Enter the string : heyyy
heyyy
heyyy
heyyy
heyyy
heyyy
heyyy
heyyy
heyyy
heyyy
-----
Process exited after 8.814 seconds with return value 0
Press any key to continue . . .
```

Compilation results...

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS print.exe
- Output Size: 128.6015625 KiB
- Compilation Time: 0.31s
```


9) PRINT NUMBERS FROM 1 TO 10



The screenshot shows the Dev-C++ IDE with a C++ program in the main editor window. The program is titled "nested ifelse big among three.cpp". The code is as follows:

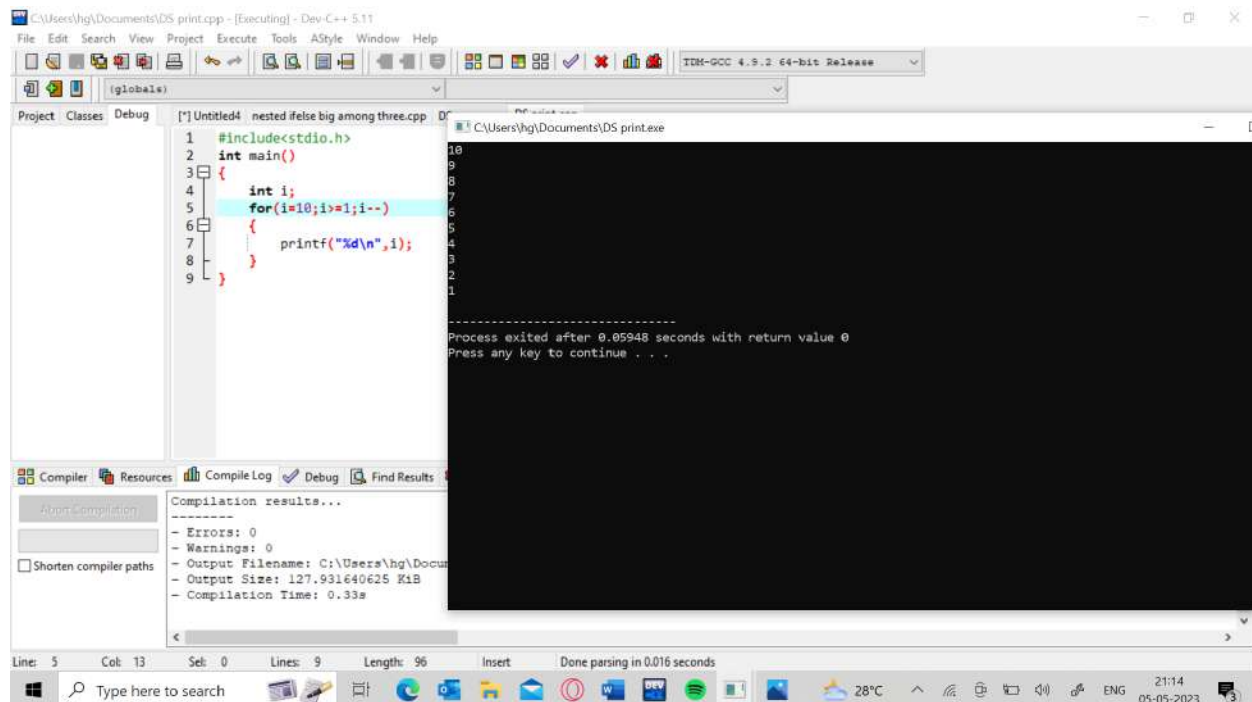
```
1 #include<stdio.h>
2 int main()
3 {
4     int i;
5     for(i=1;i<=10;i++)
6     {
7         printf("%d\n",i);
8     }
9 }
```

The code is compiled and executed, and the output window shows the numbers 1 through 10 printed on separate lines. The compilation results are shown in the bottom panel:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS print.exe
- Output Size: 127.931640625 KiB
- Compilation Time: 0.33s
```

The status bar at the bottom indicates the current line is 5, column is 16, and the file is 9 lines long with a length of 96 characters.

10) PRINT NUMBERS FROM 10 TO 1



The screenshot shows the Dev-C++ IDE with a C++ program in the main editor window. The program is titled "nested ifelse big among three.cpp". The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int i;
5     for(i=10;i>=1;i--)
6     {
7         printf("%d\n",i);
8     }
9 }
```

The code is compiled and executed, and the output window shows the numbers 10 through 1 printed on separate lines. The compilation results are shown in the bottom panel:

```
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS print.exe
- Output Size: 127.931640625 KiB
- Compilation Time: 0.33s
```

The status bar at the bottom indicates the current line is 5, column is 13, and the file is 9 lines long with a length of 96 characters.

11) GET 5 NUMBERS FROM USER AND SUM IT USING WHILE LOOP

The screenshot shows the Dev-C++ IDE with a C++ program open. The program prompts the user to enter 5 numbers and calculates their sum using a while loop. The output window shows the user inputting 10, 20, 30, 40, and 50, resulting in a sum of 150.

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,sum=0,n[20];
5     printf("Enter the 5 numbers :\n");
6     i=0;
7     while(i<5)
8     {
9         scanf("%d",&n[i]);
10        sum=sum+n[i];
11        i++;
12    }
13    printf("The sum is : %d",sum);
14 }
15 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS while.exe
- Output Size: 128,270,507,8125 KiB
- Compilation Time: 0.33s

Process exited after 4.947 seconds with return value 0
Press any key to continue . . .

12) SUM OF DIGITS USING WHILE LOOP

The screenshot shows the Dev-C++ IDE with a C++ program open. The program prompts the user to enter a number and calculates the sum of its digits using a while loop. The output window shows the user inputting 341, resulting in a sum of digits of 8.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,d,sum=0;
5     printf("Enter the number : ");
6     scanf("%d",&n);
7     while(n>0)
8     {
9         d=n%10;
10        sum=sum+d;
11        n=n/10;
12    }
13    printf("the sum of the digits is : %d",sum);
14 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS sum dig.exe
- Output Size: 128,101,562,5 KiB
- Compilation Time: 0.28s

Process exited after 2.989 seconds with return value 0
Press any key to continue . . .

13) C PROGRAM TO PRINT FIRST 'N' NUMBERS USING DO-WHILE LOOP

The screenshot shows a C program in a code editor and its execution output. The code defines a function `main()` that initializes `i=1` and `n`. It prompts the user to "Enter the Limit : ", reads the input `n` using `scanf`, and then enters a `do-while` loop. Inside the loop, it prints the value of `i` and increments it by 1 until `i` is less than or equal to `n`. The output window shows the user entering 10, and the program prints numbers 1 through 10 on separate lines. The compilation results show 0 errors and 0 warnings, with the output file named `DS do while.exe`.

```
1 #include<stdio.h>
2 int main()
3 {
4     int i=1,n;
5     printf("Enter the Limit : ");
6     scanf("%d",&n);
7     do
8     {
9         printf("%d\n",i);
10        i++;
11    }
12    while(i<=n);
13 }
14
```

Enter the Limit : 10
1
2
3
4
5
6
7
8
9
10
.....
Process exited after 0.8068 seconds with return value 0
Press any key to continue . . .

Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS do while.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.28s

14) INITIALIZE ARRAY WITH 10 ELEMENTS AND PRINT THE SAME

The screenshot shows a C program in a code editor and its execution output. The code defines a function `main()` that prompts the user to "Enter the number of elements you want to add : ", reads the input `n` using `scanf`, and then prompts the user to "Enter the elements in the array : ". It then enters a `for` loop to read 10 elements into an array `a` of size 20. After the loop, it prints the elements of the array. The output window shows the user entering 10, and then entering 10 elements (1 through 10). The program prints the same 10 elements. The compilation results show 0 errors and 0 warnings, with the output file named `DS array.exe`.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,a[20],i;
5     printf("Enter the number of elements you want to add : ");
6     scanf("%d",&n);
7     printf("Enter the elements in the array : \n");
8     for(i=0;i<n;i++)
9     {
10        scanf("%d",&a[i]);
11    }
12    printf("\n");
13    for(i=0;i<n;i++)
14    {
15        printf("%d\n",a[i]);
16    }
17 }
```

Enter the number of elements you want to add : 10
Enter the elements in the array :
1
2
3
4
5
6
7
8
9
10
.....
Process exited after 6.722 seconds with return value 0
Press any key to continue . . .

Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\DS array.exe
- Output Size: 128.7705078125 KiB
- Compilation Time: 0.28s

15) SUM OF ARRAY ELEMENTS

The screenshot shows the Dev-C++ IDE with a C++ program for calculating the sum of array elements. The code is as follows:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,i,a[20],sum=0;
5     printf("Enter the size of the array : ");
6     scanf("%d",&n);
7     printf("Enter the elements in the array : \n");
8     for(i=0;i<n;i++)
9     {
10         scanf("%d",&a[i]);
11         sum=sum+a[i];
12     }
13     printf("the sum of the elements in the array is : %d",sum);
14 }
```

The output window shows the following execution results:

```
Enter the elements in the array :
1
2
3
4
5
6
7
8
9
10
the sum of the elements in the array is : 55
.....
Process exited after 8.426 seconds with return value 0
Press any key to continue . . .
```

The compilation results show no errors or warnings, and the output file is named DS ar sum.exe.

16) REVERSE A NUMBER USING WHILE

The screenshot shows the Dev-C++ IDE with a C++ program for reversing a number using a while loop. The code is as follows:

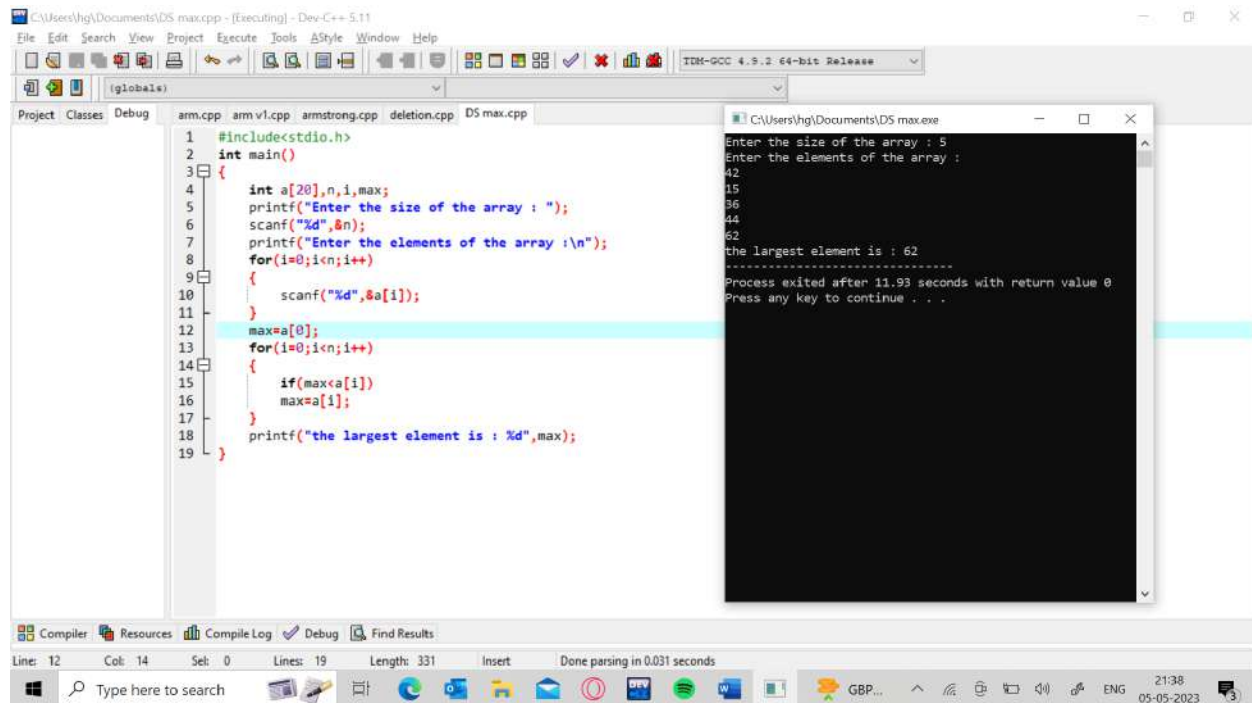
```
1 #include<stdio.h>
2 int main()
3 {
4     int n,d,rev=0;
5     printf("Enter the number : ");
6     scanf("%d",&n);
7     while(n>0)
8     {
9         d=n%10;
10        rev=rev*10+d;
11        n=n/10;
12    }
13    printf("the reverse number is : %d",rev);
14 }
```

The output window shows the following execution results:

```
the reverse number is : 54
.....
Process exited after 2.593 seconds with return value 0
Press any key to continue . . .
```

The compilation results show no errors or warnings, and the output file is named DS A1 rev.exe.

17) LARGEST ELEMENT IN AN ARRAY



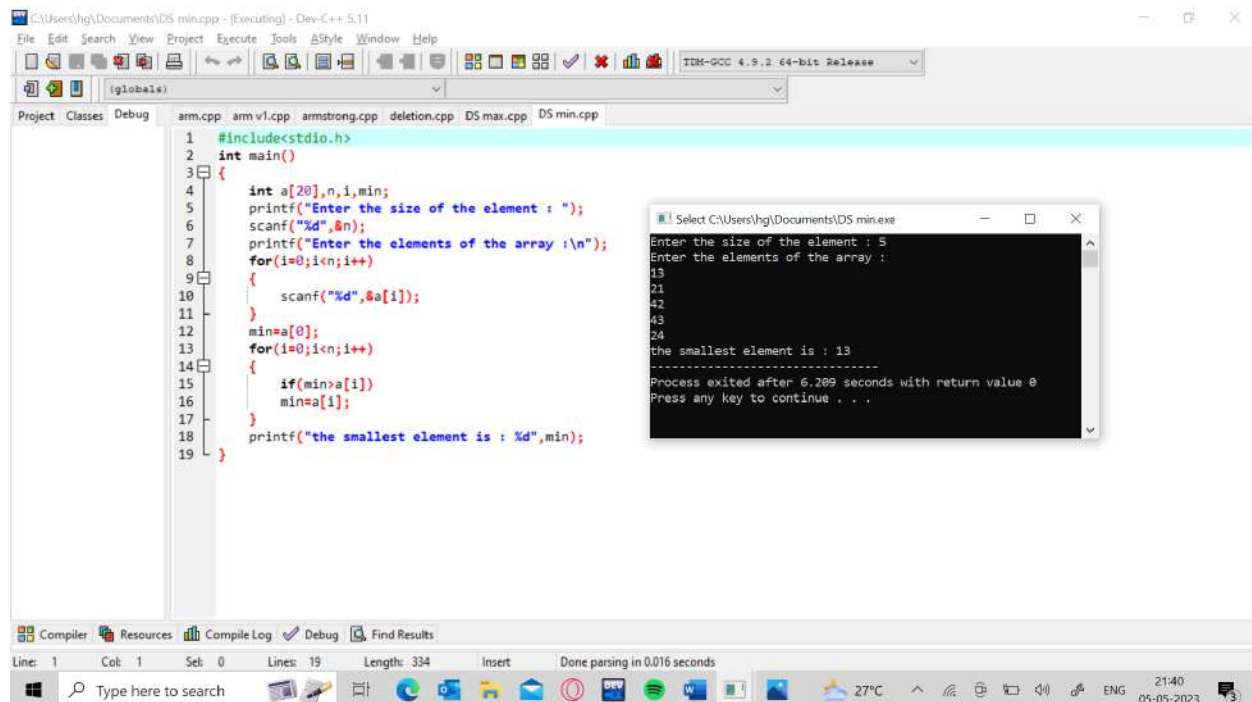
The screenshot shows the Dev-C++ IDE with a C++ program open in the editor. The program is titled 'DS max.cpp' and is located at 'C:\Users\hgh\Documents\DS max.cpp'. The code is as follows:

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

The program is executed, and the output is shown in a separate window titled 'C:\Users\hgh\Documents\DS max.exe'. The output is as follows:

```
Enter the size of the array : 5
Enter the elements of the array :
42
15
36
44
62
the largest element is : 62
.....
Process exited after 11.93 seconds with return value 0
Press any key to continue . . .
```

18)SMALLEST ELEMENT IN AN ARRAY



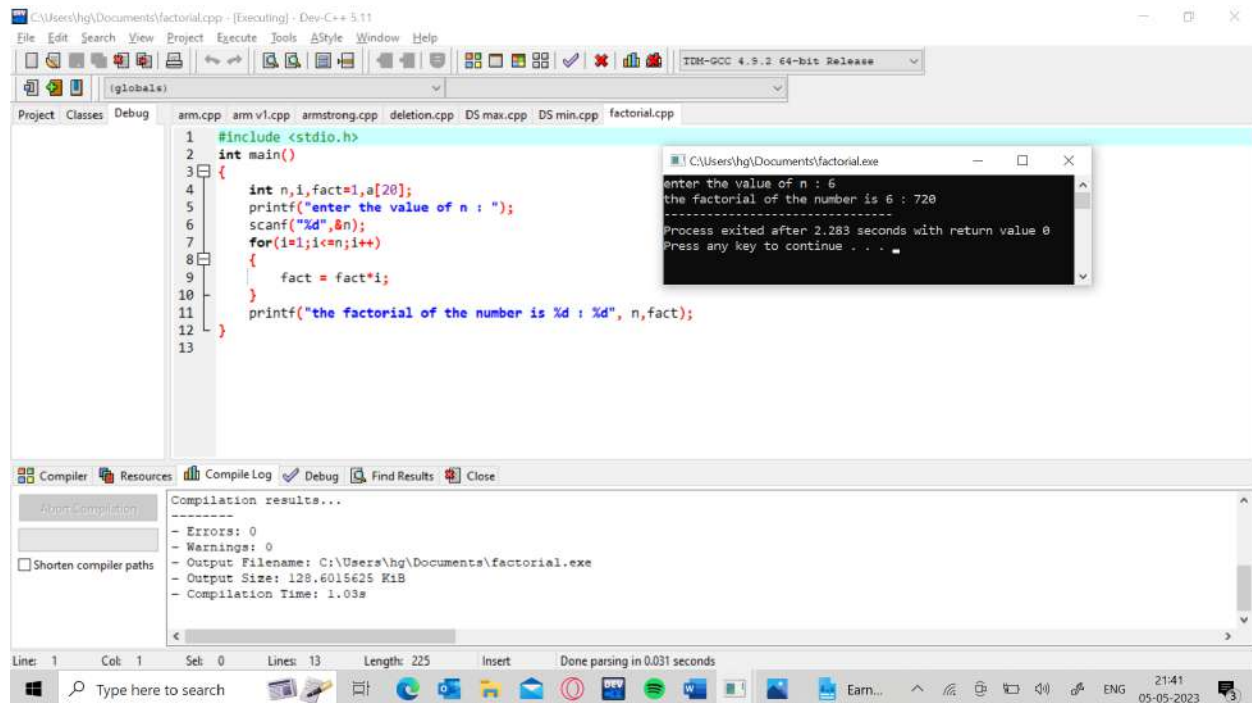
The screenshot shows the Dev-C++ IDE with a C++ program open in the editor. The program is titled 'DS min.cpp' and is located at 'C:\Users\hgh\Documents\DS min.cpp'. The code is as follows:

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

The program is executed, and the output is shown in a separate window titled 'C:\Users\hgh\Documents\DS min.exe'. The output is as follows:

```
Enter the size of the element : 5
Enter the elements of the array :
13
21
42
43
24
the smallest element is : 13
.....
Process exited after 6.209 seconds with return value 0
Press any key to continue . . .
```

19) FACTORIAL OF A NUMBER N



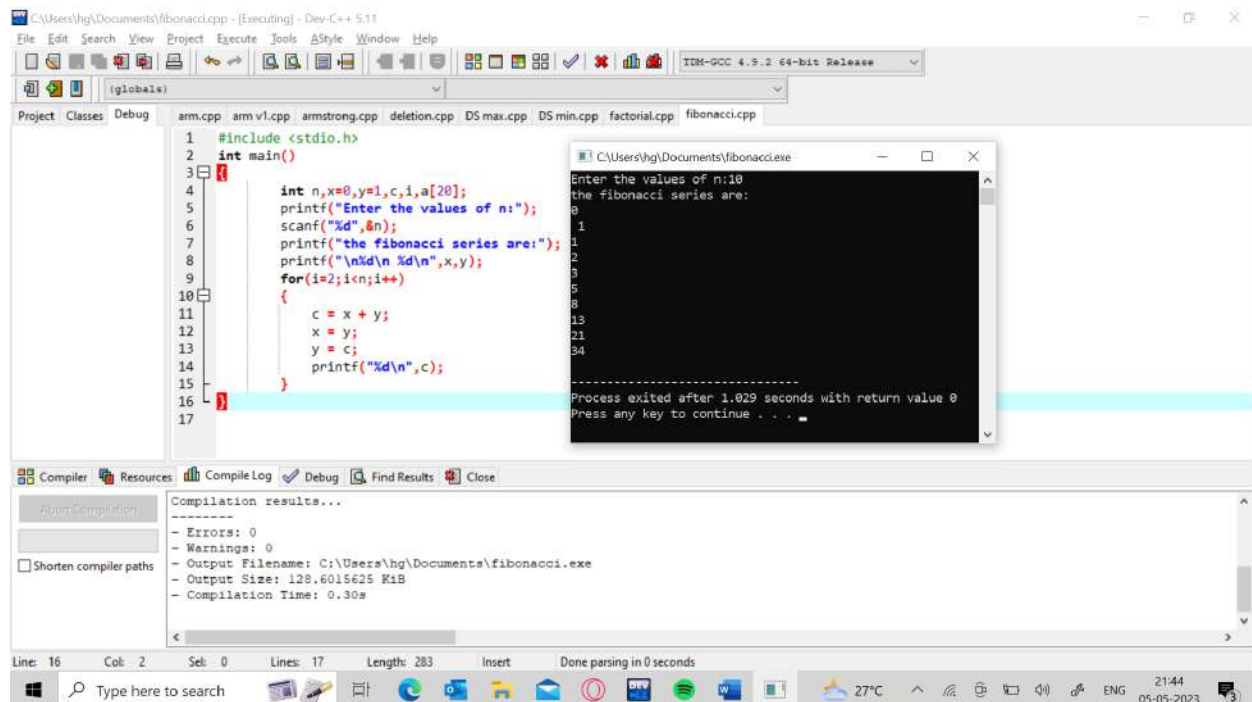
The screenshot shows the Dev-C++ IDE with the file `factorial.cpp` open. The code calculates the factorial of a number `n` entered by the user. The output window shows the user entered `6`, and the program outputs `the factorial of the number is 6 : 720`. The compilation results show 0 errors and 0 warnings.

```
1 #include <stdio.h>
2 int main()
3 {
4     int n,i,fact=1,a[20];
5     printf("enter the value of n : ");
6     scanf("%d",&n);
7     for(i=1;i<=n;i++)
8     {
9         fact = fact*i;
10    }
11    printf("the factorial of the number is %d : %d", n,fact);
12 }
13
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\factorial.exe
- Output Size: 128.6015625 KiB
- Compilation Time: 1.03s

20) FIBONACCI SERIES



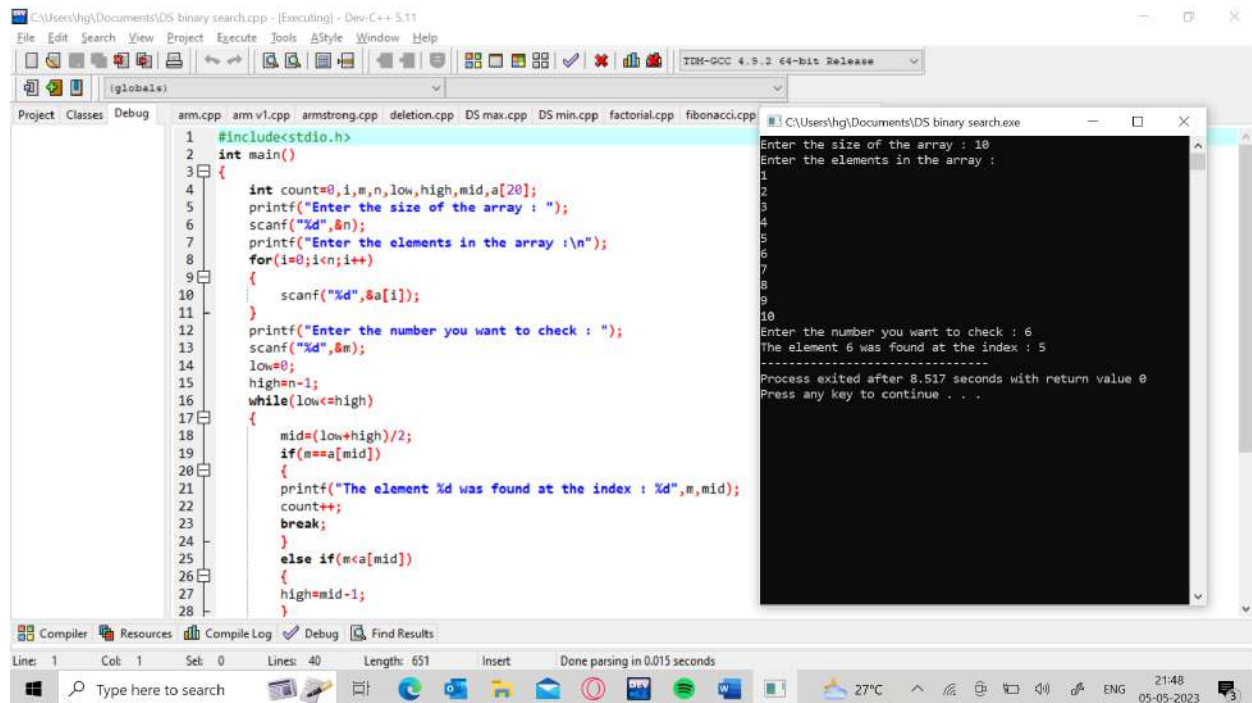
The screenshot shows the Dev-C++ IDE with the file `fibonacci.cpp` open. The code generates the Fibonacci series for a given number `n`. The output window shows the user entered `10`, and the program outputs the first 10 terms of the Fibonacci series: `0, 1, 1, 2, 3, 5, 8, 13, 21, 34`. The compilation results show 0 errors and 0 warnings.

```
1 #include <stdio.h>
2 int main()
3 {
4     int n,x=0,y=1,c,i,a[20];
5     printf("Enter the values of n:");
6     scanf("%d",&n);
7     printf("the fibonacci series are:");
8     printf("\n%d\n %d\n",x,y);
9     for(i=2;i<=n;i++)
10    {
11        c = x + y;
12        x = y;
13        y = c;
14        printf("%d\n",c);
15    }
16 }
17
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hg\Documents\fibonacci.exe
- Output Size: 128.6015625 KiB
- Compilation Time: 0.30s

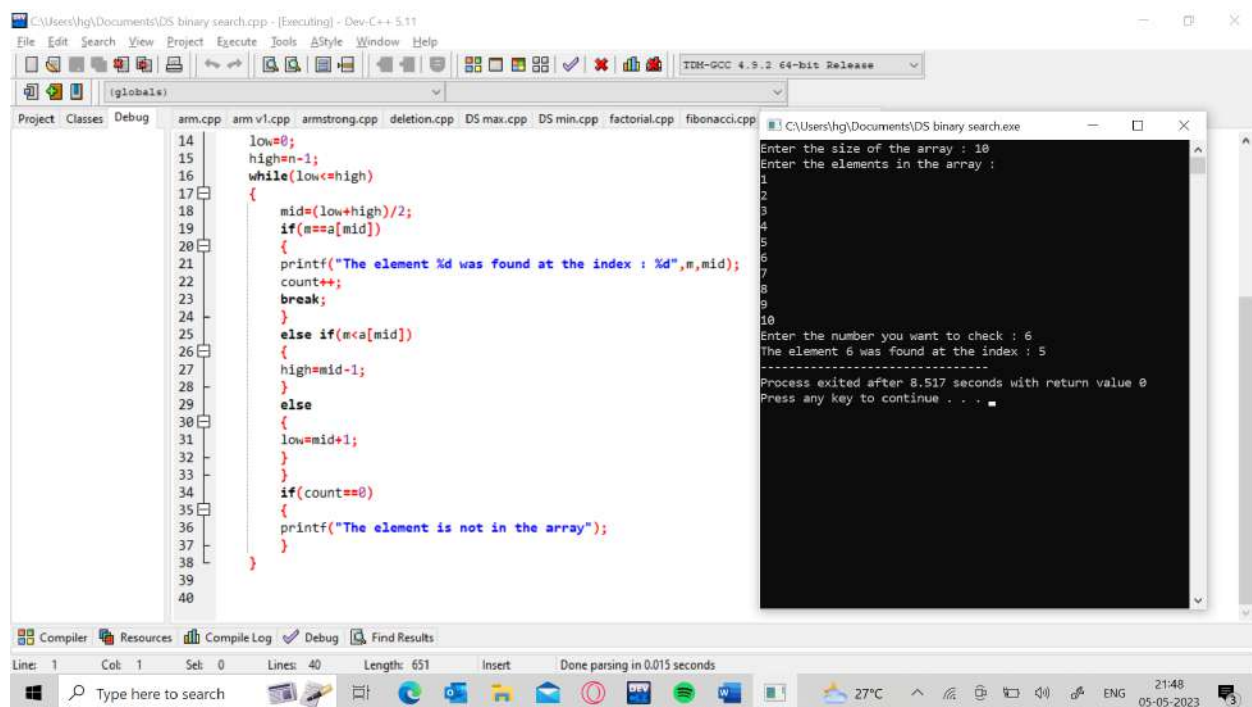
21)BINARY SEARCH IN C



```
#include<stdio.h>
int main()
{
    int count=0,i,m,n,low,high,mid,a[20];
    printf("Enter the size of the array : ");
    scanf("%d",&n);
    printf("Enter the elements in the array :\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Enter the number you want to check : ");
    scanf("%d",&m);
    low=0;
    high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(m==a[mid])
        {
            printf("The element %d was found at the index : %d",m,mid);
            count++;
            break;
        }
        else if(m<a[mid])
        {
            high=mid-1;
        }
    }
}
```

Enter the size of the array : 10
Enter the elements in the array :
1
2
3
4
5
6
7
8
9
10
Enter the number you want to check : 6
The element 6 was found at the index : 5

Process exited after 8.517 seconds with return value 0
Press any key to continue . . .

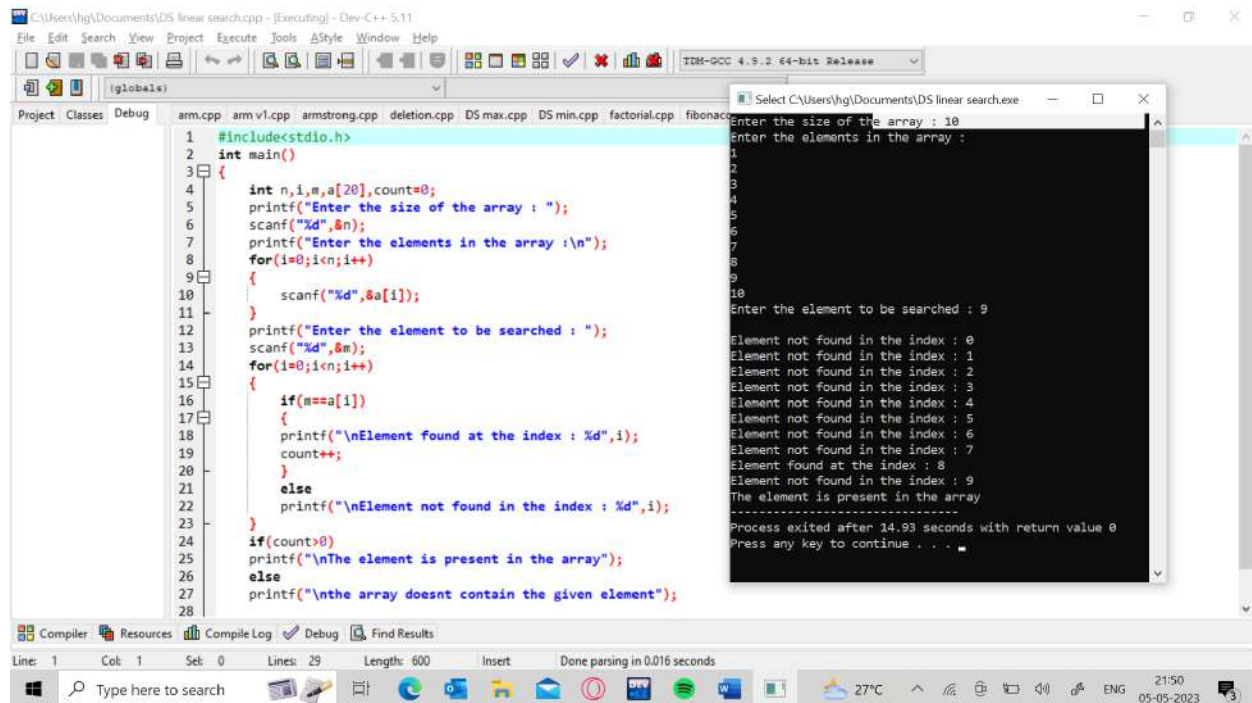


```
low=0;
high=n-1;
while(low<=high)
{
    mid=(low+high)/2;
    if(m==a[mid])
    {
        printf("The element %d was found at the index : %d",m,mid);
        count++;
        break;
    }
    else if(m<a[mid])
    {
        high=mid-1;
    }
    else
    {
        low=mid+1;
    }
    if(count==0)
    {
        printf("The element is not in the array");
    }
}
```

Enter the size of the array : 10
Enter the elements in the array :
1
2
3
4
5
6
7
8
9
10
Enter the number you want to check : 6
The element 6 was found at the index : 5

Process exited after 8.517 seconds with return value 0
Press any key to continue . . .

22) LINEAR SEARCH



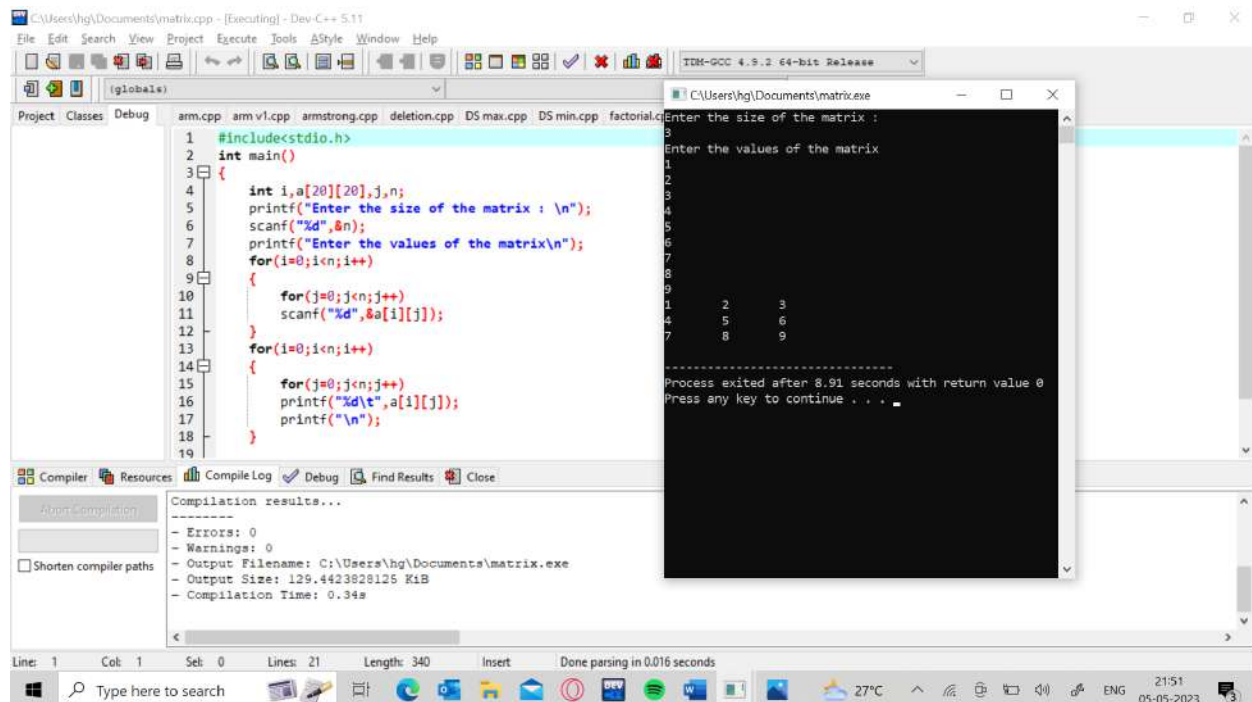
The screenshot shows a C++ IDE with a file named `linear_search.cpp`. The code implements a linear search algorithm. It prompts the user to enter the size of the array (10) and the elements (1 through 10). It then prompts for the element to be searched (9). The output shows that the element 9 is found at index 8.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,i,m,a[20],count=0;
5     printf("Enter the size of the array : ");
6     scanf("%d",&n);
7     printf("Enter the elements in the array :\n");
8     for(i=0;i<n;i++)
9     {
10         scanf("%d",&a[i]);
11     }
12     printf("Enter the element to be searched : ");
13     scanf("%d",&m);
14     for(i=0;i<n;i++)
15     {
16         if(m==a[i])
17         {
18             printf("\nElement found at the index : %d",i);
19             count++;
20         }
21         else
22             printf("\nElement not found in the index : %d",i);
23     }
24     if(count>0)
25         printf("\nThe element is present in the array");
26     else
27         printf("\nthe array doesnt contain the given element");
28 }
```

Execution output:

```
Enter the size of the array : 10
Enter the elements in the array :
1
2
3
4
5
6
7
8
9
10
Enter the element to be searched : 9
Element not found in the index : 0
Element not found in the index : 1
Element not found in the index : 2
Element not found in the index : 3
Element not found in the index : 4
Element not found in the index : 5
Element not found in the index : 6
Element not found in the index : 7
Element found at the index : 8
Element not found in the index : 9
The element is present in the array
-----
Process exited after 14.93 seconds with return value 0
Press any key to continue . . .
```

23) MATRIX IN C



The screenshot shows a C++ IDE with a file named `matrix.cpp`. The code implements a program to input and display a 3x3 matrix. It prompts the user to enter the size of the matrix (3) and the values of the matrix (1 through 9). The output displays the matrix in a 3x3 grid.

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

Execution output:

```
Enter the size of the matrix :
3
Enter the values of the matrix
1
2
3
4
5
6
7
8
9
-----
Process exited after 8.91 seconds with return value 0
Press any key to continue . . .
```

Compilation results:

```
-----
Errors: 0
Warnings: 0
Output Filename: C:\Users\hg\Documents\matrix.exe
Output Size: 129.4423828125 K1B
Compilation Time: 0.34s
```


24) POINTER IN C

```
#include<stdio.h>
int main()
{
    int n,a[n],top=-1,ch,m,i;
    printf("Enter the size of the array : ");
    scanf("%d",&n);
    while(1)
    {
        printf("\nEnter your choice : ");
        printf("\n1.PUSH \n2.POP \n3.DISPLAY \n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:
                if(top==n-1)
                {
                    printf("you cant push elements...overflow\n");
                }
                else
                {
                    top++;
                    printf("Enter the value : ");
                    scanf("%d",&m);
                    a[top]=m;
                }
                break;
            case 2:
                if(top>=-1)
                {
                    printf("The popped element is = %d",a[top]);
                    top--;
                }
                break;
            case 3:
                if(top>=-1)
                {
                    printf("Theres no element available in the array\n");
                }
                else
                {
                    for(i=0;i<=top;i++)
                    {
                        printf("\t%d\n",a[i]);
                    }
                }
                break;
        }
        printf("\nDo you wish to continue \n1.Break \n2.continue : ");
        scanf("%d",&ch);
        if(ch==1)
        {
            break;
        }
    }
}
```

Enter the size of the array : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
1
Enter the value : 10
Do you wish to continue
1.Break
2.continue : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
1
Enter the value : 20
Do you wish to continue
1.Break
2.continue : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
1
you cant push elements...overflow
Do you wish to continue
1.Break
2.continue : 2
Enter your choice :

```
break;
case 2:
{
    if(top>=-1)
    {
        printf("deletion cannot be performed further...underflow\n");
    }
    else
    {
        printf("The popped element is = %d",a[top]);
        top--;
    }
    break;
}
case 3:
if(top>=-1)
{
    printf("Theres no element available in the array\n");
}
else
{
    for(i=0;i<=top;i++)
    {
        printf("\t%d\n",a[i]);
    }
}
break;
printf("\nDo you wish to continue \n1.Break \n2.continue : ");
scanf("%d",&ch);
if(ch==1)
{
    break;
}
```

Do you wish to continue
1.Break
2.continue : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
3
10
20
Do you wish to continue
1.Break
2.continue : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
2
The popped element is = 20
Do you wish to continue
1.Break
2.continue : 2
Enter your choice :
1.PUSH
2.POP
3.DISPLAY
3
10
Do you wish to continue
1.Break

25)DOUBLE POINTER

The image displays two screenshots of a Dev-C++ IDE showing the implementation of a double pointer array. The first screenshot shows the initial setup and the first insertion. The second screenshot shows the deletion of an element and the subsequent shifting of the remaining elements.

First Screenshot:

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,a[n],top=-1,ch,m,i,bot=-1;
5     printf("Enter the size of the array : ");
6     scanf("%d",&n);
7     while(1)
8     {
9         printf("\n1.Insert the elements in the array \n2.Delete elements in array \n3.Display the contents of the array\n");
10        printf("\nEnter your choice : ");
11        scanf("%d",&ch);
12        switch(ch)
13        {
14            case 1:
15                if(bot==n-1)
16                {
17                    printf("you cant push elements....overflow\n");
18                    printf("\n");
19                }
20                else if(bot==-1)
21                {
22                    top=0;
23                    bot=0;
24                    printf("Enter the value : ");
25                    scanf("%d",&m);
26                    a[bot]=m;
27                }
28                else
29                {
30                    bot++;
31                    printf("Enter the value : ");
32                    scanf("%d",&m);
33                    a[bot]=m;
34                }
35                break;
36            case 2:
37                if(top==-1)
38                {
39                    printf("deletion cannot be performed further....underflow error\n");
40                    printf("\n");
41                }
42                else
43                {
44                    printf("The popped element is = %d",a[top]);
45                    printf("\n");
46                    top=top+1;
47                }
48                break;
49            case 3:
50                if(top==-1)
51                {
52                    printf("Theres no element available in the array\n");
53                    printf("\n");
54                }
55                else
56                {
57                    for(i=bot;i>=top;i--)
58                    {
59                        printf("\t%d\n",a[i]);
60                    }
61                    printf("\n");
62                }
63                break;
64        }
65    }
66 }
```

Second Screenshot:

```
34 a[bot]=m;
35 break;
36 case 2:
37 {
38     if(top==-1)
39     {
40         printf("deletion cannot be performed further....underflow error\n");
41         printf("\n");
42     }
43     else
44     {
45         printf("The popped element is = %d",a[top]);
46         printf("\n");
47         top=top+1;
48     }
49     break;
50 }
51 case 3:
52 if(top==-1)
53 {
54     printf("Theres no element available in the array\n");
55     printf("\n");
56 }
57 else
58 {
59     for(i=bot;i>=top;i--)
60     {
61         printf("\t%d\n",a[i]);
62     }
63     printf("\n");
64 }
65 break;
66 }
```

The output window shows the following sequence of operations:

```
Enter the size of the array : 3
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 1
Enter the value : 10
Do you wish to continue
1.Break
2.continue : 2
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 1
Enter the value : 20
Do you wish to continue
1.Break
2.continue : 2
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 1
Enter the value : 30
Do you wish to continue
1.Break
2.continue : 2
2.continue : 2
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 1
you cant push elements....overflow
Do you wish to continue
1.Break
2.continue : 2
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 3
30
20
10
Do you wish to continue
1.Break
2.continue : 2
1.Insert the elements in the array
2.Delete elements in array
3.Display the contents of the array
Enter your choice : 2
The popped element is = 10
```

