

## CLASS WORK PROGRAMS

### 1. C Program to Reverse a Number Using FOR Loop.

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays the following C program:

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int num,r,sum=0,t;
6
7     printf("Input a number: ");
8     scanf("%d",&num);
9
10    for(t=num;num!=0;num=num/10){
11        r=num % 10;
12        sum=sum*10+r;
13    }
14    printf("The number in reverse order is : %d \n",sum);
15 }
```

To the right of the code editor is a terminal window titled "C:\Users\Del\Documents\class work-1.exe". It shows the output of the program when run with the input "79". The output is:

```
Input a number: 79
The number in reverse order is : 978

Process exited after 3.08 seconds with return value 37
Press any key to continue . . .
```

Below the terminal window, the "Compiler" tab of the status bar shows compilation results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Del\Documents\class work-1.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.73s
```

The status bar at the bottom of the IDE shows the current line (Line: 14), column (Col: 2), selection (Sel: 0), lines (Lines: 14), length (Length: 266), and the message "Done parsing in 0.031 seconds".

### 2. Write a program in C such that the program will read the elements of a one-dimensional array, then compares the elements and finds which the largest two elements are in a given array.

```

C:\Users\gnane\OneDrive\Documents\2nd ques.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling

Project Classes Debug /* reversein.c.cpp errors1.cpp error2.cpp error3.cpp error4.cpp Untitled6.cpp error11.cpp Untitled22.cpp Untitled23.cpp 2nd ques.cpp
main() int
1 #include <stdio.h>
2 int main ()
3 {
4     int n = 0, i = 0, largest1 = 0, largest2 = 0, temp = 0;
5     printf ("Enter the size of the array\n");
6     scanf ("%d", &n);
7     int array[n];
8     printf ("Enter the elements\n");
9     for (i = 0; i < n; i++)
10    {
11        scanf ("%d", &array[i]);
12    }
13
14    printf ("The array elements are : \n");
15    for (i = 0; i < n; i++)
16    {
17        printf ("%d\t", array[i]);
18    }
19
20    printf ("\n");
21
22    largest1 = array[0];
23    largest2 = array[1];
25
26    if (largest1 < largest2)
27    {
28        temp = largest1;
29        largest1 = largest2;
30        largest2 = temp;
31    }
32
33    for (i = 2; i < n; i++)
34    {
35        if (array[i] > largest1)
36        {
37            largest2 = largest1;
38            largest1 = array[i];
39        }
40        else if (array[i] > largest2)
41        {
42            largest2 = array[i];
43        }
44    }
45
46    printf ("The FIRST LARGEST = %d\n", largest1);
47    printf ("THE SECOND LARGEST = %d\n", largest2);
48
49    return 0;
50 }

Compiler Resources Compile Log Debug Find Results Close
Abort Compilation Compilation results...
-----  

- Errors: 0  

- Warnings: 0  

- Output Filenames: C:\Users\gnane\OneDrive\Documents\2nd ques.exe  

- Output Size: 153.412109375 Kib  

- Compilation Time: 0.17s

```

3. kth smallest element is the minimum possible n such that there are at least k elements in the array  $\leq n$ . In other words, if the array A was sorted, then  $A[k - 1]$  ( k is 1 based, while the arrays are 0 based )C Program for finding the desired kth smallest element in an array.

```

C:\Users\gnane\OneDrive\Documents\3rd ques.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Profiling

Project Classes Debug /* Untitled: 3rd ques.cpp
main() int
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Compare function for qsort
5 int cmpfunc(const void* a, const void* b)
6 {
7     return (* (int*)a) - (* (int*)b);
8 }
9
10 // Function to return K'th smallest
11 // element in a given array
12 int kthSmallest(int arr[], int N, int K)
13 {
14     // Sort the given array
15     qsort(arr, N, sizeof(int), cmpfunc);
16
17     // Return K'th element in the sorted array
18     return arr[K - 1];
19 }
20
21 // Driver's code
22 int main()
23 {
24     int arr[] = { 12, 3, 5, 7, 19 };
25     int N = sizeof(arr) / sizeof(arr[0]), K = 2;
26
27     // Function call
28     printf ("K'th smallest element is %d", kthSmallest(arr, N, K));
29     return 0;
30 }

Compiler Resources Compile Log Debug Find Results Close
Abort Compilation Compilation results...
-----  

- Errors: 0  

- Warnings: 0  

- Output Filenames: C:\Users\gnane\OneDrive\Documents\3rd ques.exe  

- Output Size: 153.142578125 Kib  

- Compilation Time: 0.20s

```

4. Write a program to search the given element using binary search method and display its position in a linear array.

Sample Input;

Array of elements = {28, 12, 15, 122, 10, 33, 11}

Element to search = 15

SampleOutput:

Given element 23 is found at 3 rd position

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a C program named '4th qua.cpp'. The code implements a linear search algorithm. It defines a function 'linearSearch' that takes an array and a value, then iterates through the array to find the value. If found, it returns the index; if not found, it returns -1. The main function initializes an array with values {28, 40, 30, 11, 57, 41, 25, 14, 52}, sets the value to search for as 41, and calls the search function. It then prints the array elements and the result of the search. The output window on the right shows the execution results: the array elements are listed as 28 40 30 11 57 41 25 14 52, the element to search for is 41, and the message 'Element is present at 6 position of array'. Below the output window, the compiler log shows that there were no errors or warnings, and the compilation was successful.

```
#include <stdio.h>
int linearSearch(int a[], int n, int val) {
    // Going through array sequentially
    for (int i = 0; i < n; i++) {
        if (a[i] == val)
            return i+1;
    }
    return -1;
}
int main() {
    int a[] = {28, 40, 30, 11, 57, 41, 25, 14, 52}; // given array
    int val = 41; // value to be searched
    int n = sizeof(a) / sizeof(a[0]); // size of array
    int res = linearSearch(a, n, val); // Store result
    printf("The elements of the array are - ");
    for (int i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\nElement to be searched is - %d", val);
    if (res == -1)
        printf("\nElement is not present in the array");
    else
        printf("\nElement is present at %d position of array", res);
    return 0;
}
```

5. C Program to Find Sum of Natural Numbers Using While Loop

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named '5 quas.cpp' with the following code:

```
1 #include <stdio.h>
2 #include <conio.h>
3 int main()
4 {
5     int num, i, sum = 0; // declare local variables
6     printf(" Enter a positive number: ");
7     scanf("%d", &num);
8     for (i = 0; i <= num; i++)
9     {
10         sum = sum + i;
11     }
12     // display the sum of natural number
13     printf("\n Sum of the first %d number is: %d", num, sum);
14     getch();
15 }
```

To the right of the code editor is a terminal window titled 'C:\Users\gnane\OneDrive\Documents\5 quas.exe'. It shows the output of the program:

```
Enter a positive number: 89
Sum of the first 89 number is: 4009.
Process exited after 0.126 seconds with return value 0
Press any key to continue . . .
```

The bottom panel of the IDE shows the compiler results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\5 quas.exe
- Output Size: 153.2412109375 KiB
- Compilation Time: 0.24s
```

## 6. C Program to Find GCD of two Numbers Using For Loop

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named '6th qua.cpp' with the following code:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int Num1, Num2, i, GCD;
6
7     printf("Please Enter two integer Values \n");
8     scanf("%d %d", &Num1, &Num2);
9
10    for(i = 1; i <= Num1 && i <= Num2; i++)
11    {
12        if(Num1 % i == 0 && Num2 % i == 0)
13        {
14            GCD = i;
15        }
16
17        printf("GCD of %d and %d is = %d", Num1, Num2, GCD);
18    }
19
20    return 0;
21 }
```

To the right of the code editor is a terminal window titled 'C:\Users\gnane\OneDrive\Documents\6th qua.exe'. It shows the output of the program:

```
Please Enter two integer Values
34 67
GCD of 34 and 67 is = 1
Process exited after 6.893 seconds with return value 0
Press any key to continue . . .
```

The bottom panel of the IDE shows the compiler results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\6th qua.exe
- Output Size: 153.240234375 KiB
- Compilation Time: 0.17s
```

## 7. Predict the output or error(s) for the following program

```

main()
{
int i=5;
printf("%d%d%d%d%d",i++,i--,++i,--i,i);
}

```

The screenshot shows the Dev-C++ IDE interface. The code editor displays the following C program:

```

1 #include<stdio.h>
2 int main()
3 {
4     int i=5;
5     printf("%d%d%d%d%d",i++,i--,++i,--i,i);
6     return 0;
7 }
8

```

The terminal window shows the output of the program:

```

Select C:\Users\gnane\OneDrive\Documents\errors 1.exe
Process exited after 0.2008 seconds with return value 0
Press any key to continue . . .

```

The status bar at the bottom shows compilation results:

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\errors 1.exe
- Output Size: 152.9013671875 KiB
- Compilation Time: 0.17s

### 8. Predict the output or error(s) for the following program

```

main()
{
char s[ ]="man";
int i;
for(i=0;s[ i ];i++)
printf("\n%c%c%c%c",s[ i ],*(s+i),*(i+s),i[s]);
}

```

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named 'error 2.cpp' with the following code:

```
1 #include<stdio.h>
2 main()
3 {
4     char s[]="man";
5     int i;
6     for(i=0;s[ i ];i++)
7         printf("\n%d%d%d%d",s[ i ],*(s+i),*(i+s),i[s]);
8     return 0;
9 }
10
```

To the right of the code editor is a terminal window titled 'C:\Users\gnane\OneDrive\Documents\error 2.exe' showing the program's output:

```
10100100100
07970797
100100100100
Process exited after 0.1896 seconds with return value 0
Press any key to continue . . .
```

Below the code editor is a 'Compiler' tab with the following compilation results:

```
Compilation results...
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\error 2.exe
- Output Size: 152.9013671875 KiB
- Compilation Time: 0.17s
```

The status bar at the bottom shows the date and time: 23-08-2023 12:31.

9. Predict the output or error(s) for the following program

```
#include<stdio.h>

main()
{
    char s[]={‘a’,’b’,’c’,\n’,’c’,\0’};
    char *p,*str,*str1;
    p=&s[3];

    str=p;
    str1=s;
    printf("%d",++*p + ++*str1-32);
}
```

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named 'error 3.cpp' with the following code:

```
1 #include<stdio.h>
2 main()
3 {
4     char s[]={ 'a','b','c','\n','c','\0'};
5     char *p,*str,*str1;
6     p=&s[3];
7
8     str=p;
9     str1=s;
10    printf("%d",++*p + ++*str1-32);
11    return 0;
12 }
13
```

To the right of the code editor is a terminal window showing the execution results:

```
C:\Users\gnane\OneDrive\Documents\error 3.exe
Process exited after 0.1731 seconds with return value 0
PRESS ANY KEY TO CONTINUE . . .
```

Below the code editor is the Compiler Log window, which shows the following compilation results:

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\error 3.exe
- Output Size: 152.9013671875 Kib
- Compilation Time: 0.17s

The system tray at the bottom right indicates the date and time as 23-03-2023 12:36.

10. Predict the output or error(s) for the following program

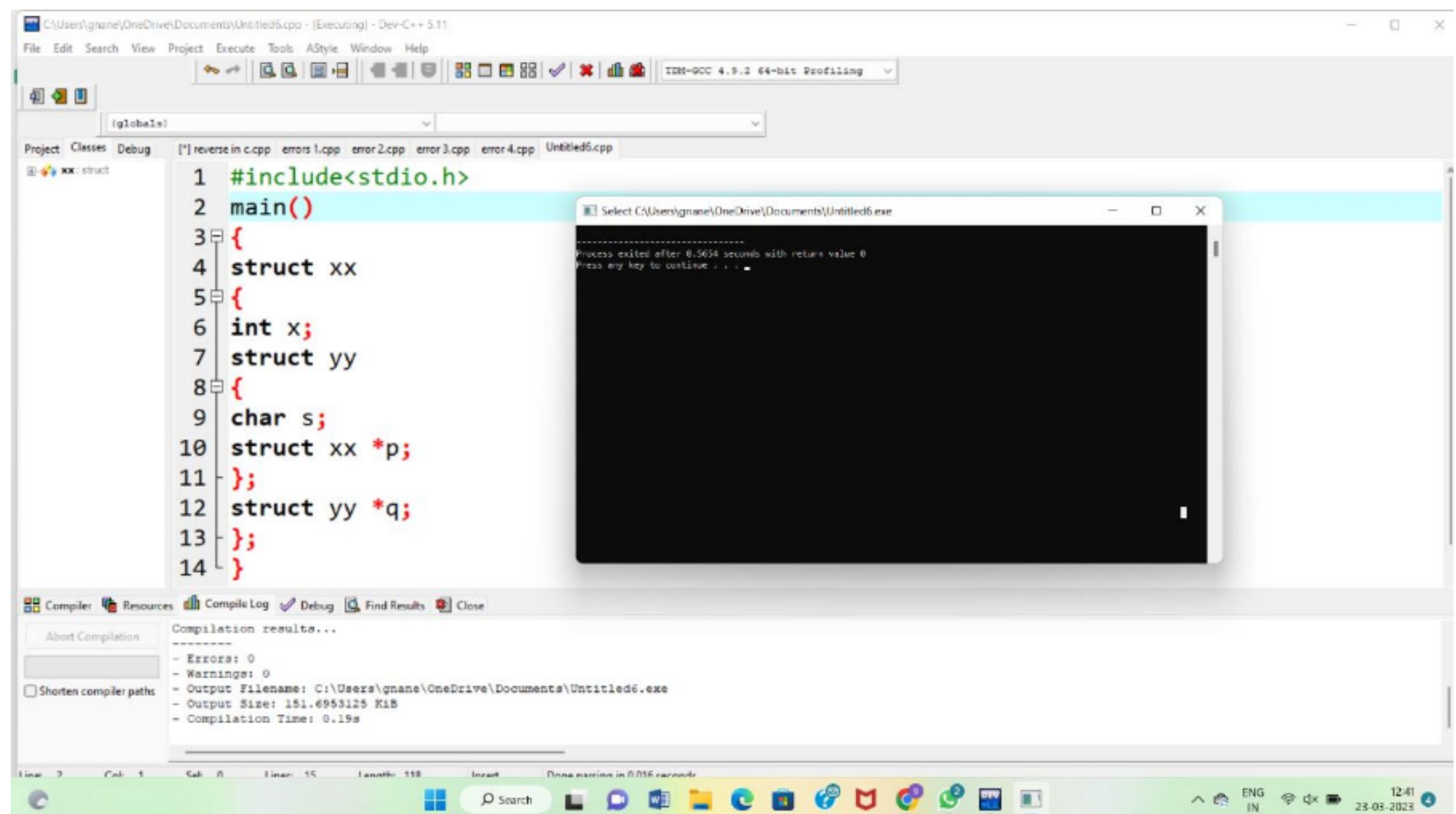
```
main()

{

struct xx
{
int x;
struct yy
{
char s;
struct xx *p;
};

struct yy *q;
};

}
```



11. Predict the output or error(s) for the following program

```
main()
{
char *p;
p="Hello";
printf("%c\n",*&p);
}
```

The screenshot shows the Dev-C++ IDE interface. The main window displays a C code file named 'error 11.cpp'. The code prints the string "Hello" to the console. The output window shows the execution results, including a warning message about deprecated conversion from string constant to 'char\*'. The taskbar at the bottom indicates the system status.

```

1 #include<stdio.h>
2 main()
3 {
4     char *p;
5     p=("Hello");
6     printf("%c\n",*p);
7     return 0;
8 }
9

```

12. Write a program to enter the marks of n student in four subjects using structure. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

**Sample Input & Output:**

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

**Test cases:**

1. 18, 76, 93, 65
2. 73, 78, 79, 75
3. 98, 106, 120, -95
4. 96, 73, AA, 95

## 5. 78,59.8,\*\*,79

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named '12 th qu.cpp' with code for calculating grades based on marks. The output window shows the program's execution results.

```
#include<stdio.h>
int main()
{
    int marks;
    printf("Enter your marks ");
    scanf("%d",&marks);
    if(marks<0 || marks>100)
    {
        printf("Wrong Entry");
    }
    else if(marks<50)
    {
        printf("Grade F");
    }
    else if(marks>=50 && marks<60)
    {
        printf("Grade D");
    }
    else if(marks>=60 && marks<70)
    {
        printf("Grade C");
    }
    else if(marks>=70 && marks<80)
    {
        printf("Grade B");
    }
}
```

Output window content:

```
C:\Users\gnane\OneDrive\Documents\12 th qu.exe
Enter your marks 98
Grade A
Process exited after 7.736 seconds with return value 0
Press any key to continue . . .
```

## 13. Print the following hill pattern

A B C D E D C B A

A B C D C B A

A B C B A

A B A

A

C:\Users\gnane\OneDrive\Documents\13 th.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Profiling

(globals)

Project Classes Debug 12th quc.cpp 13 th.cpp

```

1 #include <stdio.h>
2 int main() {
3     int i, j, rows;
4     printf("Enter the number of rows: ");
5     scanf("%d", &rows);
6     for (i = 1; i <= rows; ++i) {
7         for (j = 1; j <= i; ++j) {
8             printf("* ");
9         }
10        printf("\n");
11    }
12    return 0;
13 }
14

```

Process exited after 4.846 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\13 th.exe
- Output Size: 153.2431640625 KiB
- Compilation Timer: 0.22s

Line: 14 Col: 1 Sel: 0 Lines: 14 Length: 267 Insert Done parsing in 0.016 seconds

## 14. Write a program for matrix multiplication?

C:\Users\gnane\OneDrive\Documents\14.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Profiling

(globals)

Project Classes Debug 12th quc.cpp 13 th.cpp 14.cpp

```

1 #include<stdio.h>
2 #include<stdlib.h>
4 int main()
5 {
6     int a[10][10], b[10][10], mul[10][10], r, c, i, j, k;
7     system("cls");
8     printf("enter the number of rows");
9     scanf("%d",&r);
10    printf("enter the number of columns");
11    scanf("%d",&c);
12    printf("enter the first matrix element-\n");
13    for(i=0;i<r;i++)
14    {
15        for(j=0;j<c;j++)
16        {
17            scanf("%d",&a[i][j]);
18        }
19    }
20    printf("enter the second matrix element-\n");
21    for(i=0;i<r;i++)
22    {
23        for(j=0;j<c;j++)
24        {
25            scanf("%d",&b[i][j]);
26        }
27    }
28    printf("multiply of the matrix-\n");
29    for(i=0;i<r;i++)
30    {

```

Process exited after 23.54 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\gnane\OneDrive\Documents\14.exe
- Output Size: 154.0830078125 KiB
- Compilation Timer: 0.19s

Line: 50 Col: 2 Sel: 0 Lines: 50 Length: 930 Insert Done parsing in 0 seconds

**15.** Write a Program to display the diagonal elements in a matrix array and also find the sum of them.

Sample input:

1 4 3

4 2 6

7 1 3

Output:

Diagonal Elements are 1 2 3

Sum of diagonal elements = 6

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C program named '15th ques.cpp' which reads a square matrix from the user and prints its diagonal elements along with their sum. The output window shows the execution of the program, including user input for matrix size and elements, the printed matrix, and the calculated sum of 6. The compiler log window at the bottom shows a successful compilation with no errors or warnings.

```
#include<stdio.h>
int main()
{
    int mat[12][12];
    int i,j,row,col,sum=0;
    printf("Enter the number of rows and columns for 1st matrix\n");
    scanf("%d%d",&row,&col);
    printf("Enter the elements of the matrix\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            scanf("%d",&mat[i][j]);
        }
    }
    printf("The matrix\n");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }
    //To add diagonal elements
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            if(i==j)
                sum+=mat[i][j];
        }
    }
    printf("The sum of diagonal elements of a square matrix = %d",sum);
}
```

**16.** Write a program in C to swap elements using call by reference.

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named 'work-12.c' which swaps two integers. The code uses pointers to swap values. A terminal window shows the execution of the program, entering values 7 and 9, and displaying the swapped values 9 and 7. Below the terminal is a 'Compiler' tab showing compilation results with 0 errors and 0 warnings.

```
#include<stdio.h>
void swap(int *a, int *b);
int main()
{
    int x, y;
    printf("Enter the value of x and y: ");
    scanf("%d %d",&x,&y);
    printf("Before Swapping: x = %d, y = %d\n", x, y);
    swap(&x, &y); // passing addresses of x and y to the function
    printf("After Swapping: x = %d, y = %d\n", x, y);
    return 0;
}
void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

C:\Users\DELL\Documents\class work-12.exe  
Enter the value of x and y: 7 9  
Before Swapping: x = 7, y = 9  
After Swapping: x = 9, y = 7  
-----  
Process exited after 3.73 seconds with return value 0  
Press any key to continue . . .

Compilation results...  
-----  
- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\DELL\Documents\class work-12.exe  
- Output Size: 129.119140625 KiB  
- Compilation Time: 0.58s

## 17. Write C program to Copy file contents to another file.

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named 'work-17.c' which copies the contents of 'source.txt' to 'destination.txt'. The code uses FILE pointers and fgetc/fputc functions. A terminal window shows the execution of the program, indicating an error opening files. Below the terminal is a 'Compiler' tab showing compilation results with 0 errors and 0 warnings.

```
#include <stdio.h>
int main()
{
    FILE *source_file, *destination_file;
    char ch;

    source_file = fopen("source.txt", "r");
    destination_file = fopen("destination.txt", "w");
    if (source_file == NULL || destination_file == NULL)
    {
        printf("Error opening files!");
        return 1;
    }
    while ((ch = fgetc(source_file)) != EOF)
    {
        fputc(ch, destination_file);
    }
    printf("File copied successfully!");
    fclose(source_file);
    fclose(destination_file);
    return 0;
}
```

C:\Users\DELL\Documents\class work-17.exe  
Error opening files!  
-----  
Process exited after 0.05163 seconds with return value 1  
Press any key to continue . . .

Compilation results...  
-----  
- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\DELL\Documents\class work-17.exe  
- Output Size: 129.6129046875 KiB  
- Compilation Time: 0.45s

## 18. Write a program in C to read the file and store the lines into an array.

**Test Data :**

Input the file name to be opened : test.txt

**Expected Output :**

The content of the file test.txt are :

test line 1  
test line 2  
test line 3  
test line 4

**Test cases**

Sample.txt  
sample1.txt  
S123.txt  
S@12.txt  
Sample S.txt

The screenshot shows the Dev-C++ IDE interface. The code editor displays a C++ program named 'qus.cpp' which reads lines from a file and stores them in an array. The terminal window shows the user inputting 'test.txt' and the program outputting the four lines of content. The compiler results window shows no errors or warnings.

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

#define LSIZ 128
#define RSIZ 128

int main(void)
{
    char line[RSIZ][LSIZ];
    char fname[20];
    FILE *fptr = NULL;
    int i = 0;
    int tot = 0;
    printf("\n\n Read the file and store the lines into an array :\n");
    printf("-----\n");
    printf(" Input the filename to be opened : ");
    scanf("%s", fname);

    fptr = fopen(fname, "r");
    while(fgets(line[i], LSIZ, fptr))
    {
        line[i][strlen(line[i]) - 1] = '\0';
        i++;
    }
    tot = i;
    printf("\n The content of the file %s are : \n", fname);
    for(i = 0; i < tot; +i)
    {
        printf("%s\n", line[i]);
    }
}
```

**19.** Two strings consisting binary value of two numbers. Print the sum of the numbers in binary.

Input1: s1=" 1101" , s2=" 100101"

Result: " 110010"

Input2: s1=" 11100100111101" ,s2=" 10000"

Result: " 111001010001101"

The screenshot shows the Dev-C++ IDE interface. The code editor displays a C program named '19.cpp' which adds two binary strings. The main function includes a 'sum' function that takes two character arrays and their lengths. It uses a loop to iterate through the strings, calculate the sum of bits, and handle carry. A separate loop handles any leftover characters from the first string. The output window shows the execution of the program, where it prompts for binary numbers '11' and '10', performs the addition, and outputs the result '11 + 10 = 100'.

```
#include <stdio.h>
#include <string.h>
int main()
{
    sum("11", "10");
    return 0;
}

void sum(char b1[], char b2[], int l1, int l2)
{
    int carry = 0, temp, num1, num2, i;
    char result[100];
    result[l1 + 1] = '\0';

    while (l2 > 0) {
        num1 = b1[l1 - 1] - '0';
        num2 = b2[l2 - 1] - '0';
        temp = num1 + num2 + carry;
        if (temp >= 2) {
            carry = 1;
            temp = temp % 2;
        }
        result[l1] = temp + '0';
        l1--;
        l2--;
    }

    // This loop will add leftover
    // characters of first strings.
    while (l1 - 1 >= 0) {
        temp = b1[l1 - 1] + carry - '0';
        if (temp >= 2) {
            carry = 1;
            temp = temp % 2;
        }
        result[l1] = temp + '0';
        l1--;
    }
}
```

20. Write a Program to illustrates the use of continue statement.

The screenshot shows the Dev-C++ IDE interface. The code editor displays a C program named 'class work-20.c' that prints odd numbers from 1 to 10. It uses a for loop and an if condition to check if a number is divisible by 2. If it is, the continue statement is executed, skipping the rest of the loop body for that iteration. The output window shows the execution of the program, which prints the odd numbers 1, 3, 5, 7, and 9.

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i <= 10; i++)
    {
        if (i % 2 == 0)
        {
            continue;
        }
        printf("%d\n", i);
    }
    return 0;
}
```

21. Write a program to calculate square and cube of an entered number. Use function as an argument

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C program named 'class work-21.c'. The program includes standard input-output headers and defines three functions: 'calculate', 'square', and 'cube'. The 'calculate' function prompts the user for a number, calls 'square' and 'cube' to compute the square and cube respectively, and then prints the results. The 'square' and 'cube' functions return the calculated values. The compiler log window at the bottom shows that the compilation was successful with no errors or warnings.

```
#include <stdio.h>
void calculate(int num, int (*square)(int), int (*cube)(int));
int square(int num);
int cube(int num);
int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    calculate(num, square, cube);
    return 0;
}
void calculate(int num, int (*square)(int), int (*cube)(int))
{
    int sq = square(num);
    int cb = cube(num);
    printf("The square of %d is %d\n", num, sq);
    printf("The cube of %d is %d\n", num, cb);
}
int square(int num)
{
    return num * num;
}
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DELL\Documents\class work-21.exe
- Output Size: 129.1640625 KiB
- Compilation Time: 0.47s

22. What is the output of the program given below?

```
main()
{
    signed char i=0;
    for(;i>=0;i++);
    printf(" %d\n" ,i);
}
```

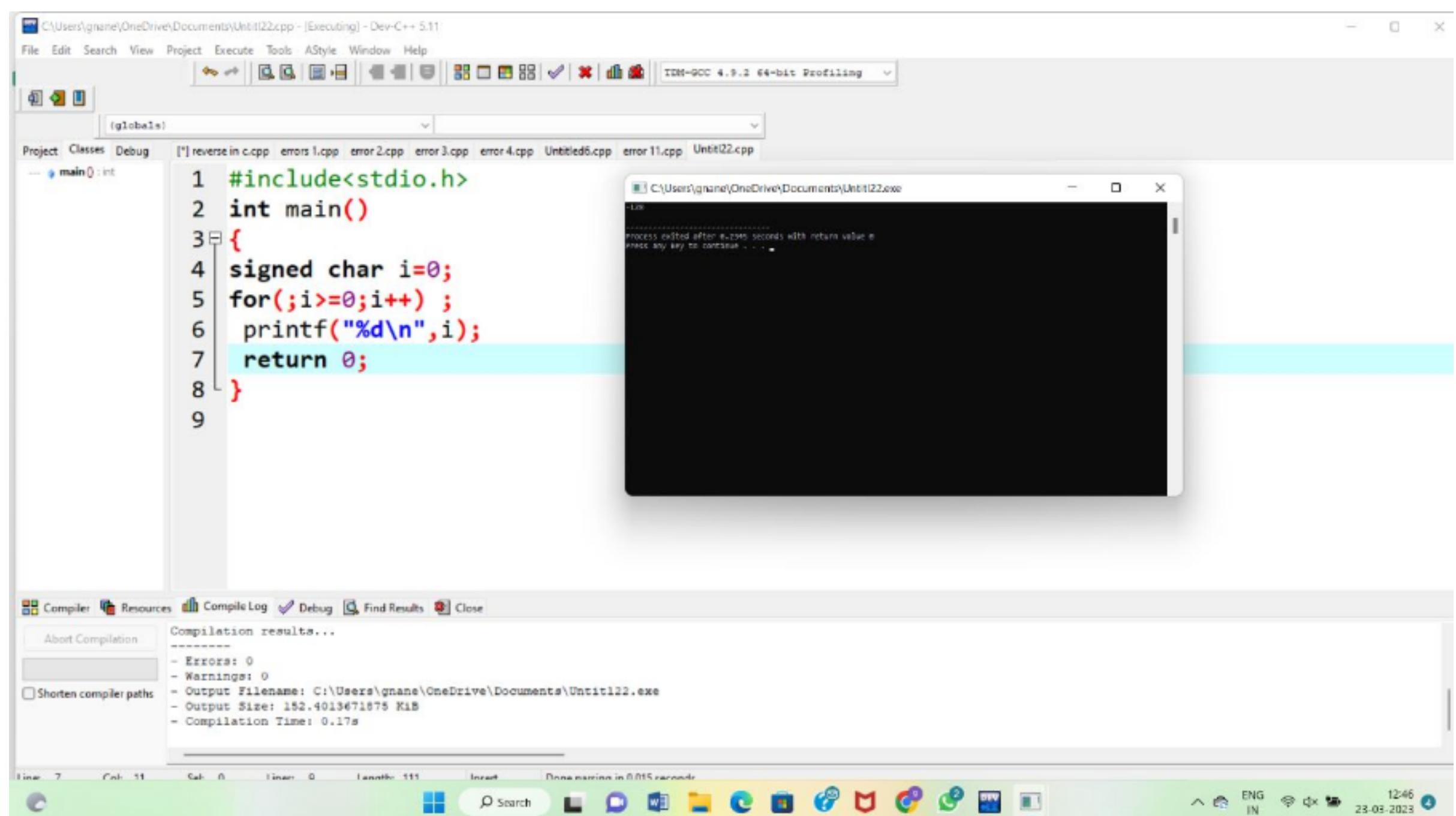
22. What is the output of the program given below?

```
int aaa()
{
    printf(" Hi" );
}
int bbb()
{
    printf(" hello" );
}
int ccc()
{
```

```

printf(" bye" );
}
main()
{
    int ( * ptr[3] ) ();
    ptr[0] = aaa;
    ptr[1] = bbb;
    ptr[2] =ccc;
    ptr[2]();
}

```



**23.** Find out the logic missing statements and display the final output.

```

#include <stdio.h>
#include<conio.h>

main()
{
    int i, n, d, sum;
    clrscr();

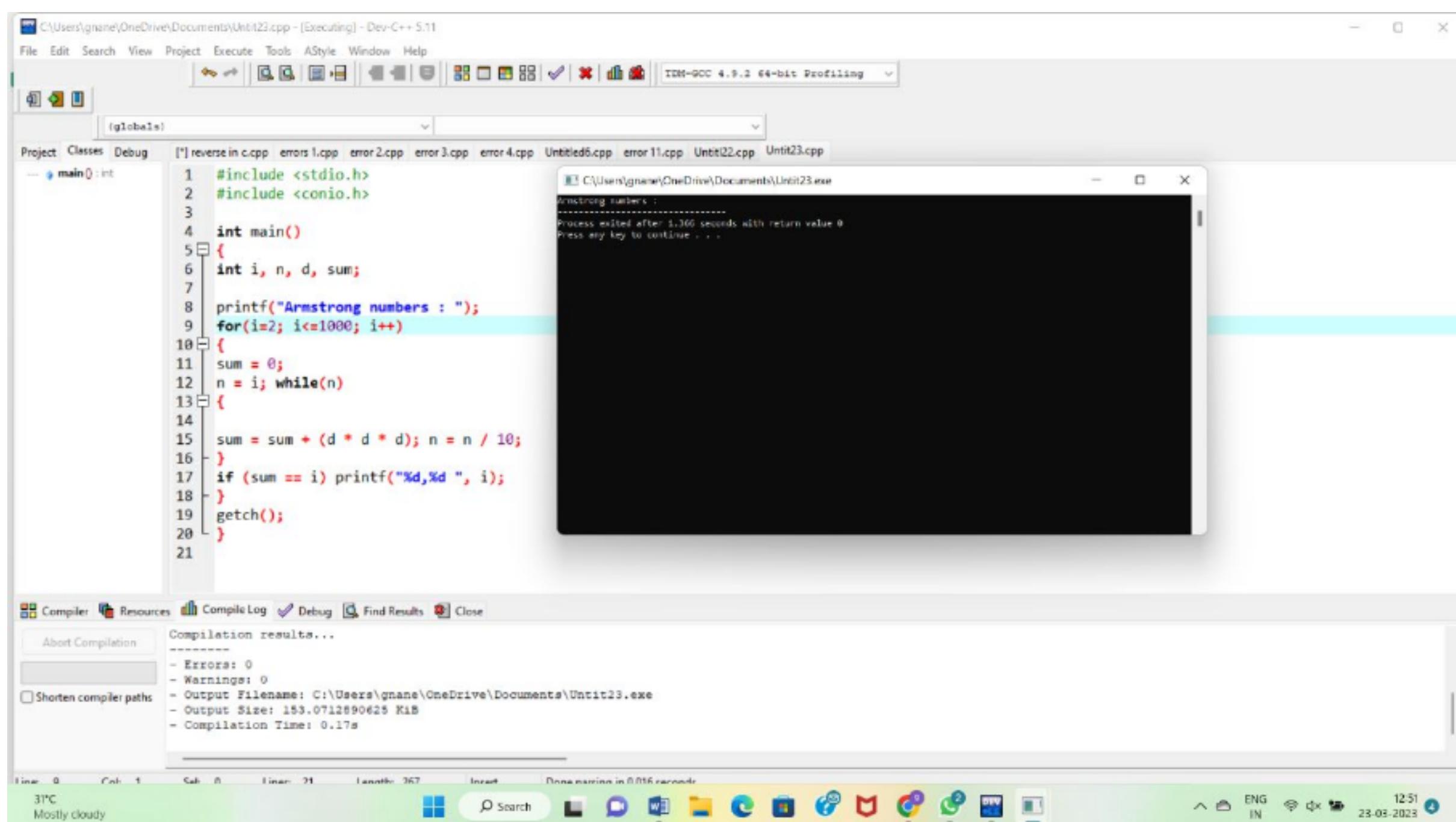
    printf("Armstrong numbers :"); for(i=2;
    i<=1000; i++)
    {

```

```

sum = 0;
n = i; while(n)
{
    sum = sum + (d * d *d); n = n /
    10;
}
if (sum == i) printf("%d ",i);
}
getch();
}

```



## 24. To generate employee payroll for an organization using structure.

1. Define *employee* structure with fields *empid*, *ename*, *basic*, *hra*, *da*, *it*, *gross* and *netpay*
2. Read number of employees *n*
3. Read *empid*, *ename*, and *basic* for *n* employees in an array of structure.
4. For each employee, compute
  - hra* = 2% of *basic*
  - da* = 1% of *basic*
  - gross* = *basic* + *hra* +
  - da it* = 5% of *basic*
  - netpay* = *gross* - *it*
5. Print *empid*, *ename*, *basic*, *hra*, *da*, *it*, *gross* and *netpay* for all employees

**25.** To implement a simple calculator using switch case statement.

Read the *operator* symbol and operands  $n1, n2$

If *operator* = + then calculate

*result* =  $n1 + n2$

Else if *operator* = – then

calculate *result* =  $n1 - n2$

Else if *operator* = \* then

calculate *result* =  $n1 * n2$

Else if *operator* = / then

calculate *result* =  $n1 / n2$

Else if *operator* = % then

calculate *result* =  $n1 \% n2$

Else

print "Invalid operator"

The screenshot shows the Dev-C++ IDE interface. The main window displays a C program named 'class work-25.c' with code for performing arithmetic operations (+, -, \*, /) on two double precision numbers. The code includes input validation for division by zero and invalid operators. The output window shows the execution of the program, where it prompts for an operator and two operands, then prints the result of the operation.

```

1 #include <stdio.h>
2
3 int main()
4 {
5     char operator;
6     double num1, num2, result;
7
8     printf("Enter an operator (+, -, *, /): ");
9     scanf("%c", &operator);
10
11    printf("Enter two operands: ");
12    scanf("%lf %lf", &num1, &num2);
13
14    switch (operator)
15    {
16        case '+':
17            result = num1 + num2;
18            printf("%.2lf + %.2lf = %.2lf", num1, num2, result);
19            break;
20
21        case '-':
22            result = num1 - num2;
23            printf("%.2lf - %.2lf = %.2lf", num1, num2, result);
24            break;
25
26        case '*':
27            result = num1 * num2;
28            printf("%.2lf * %.2lf = %.2lf", num1, num2, result);
29            break;
30
31        case '/':
32            if (num2 == 0)
33            {
34                printf("Error: Division by zero");
35            }
36            else
37            {
38                result = num1 / num2;
39                printf("%.2lf / %.2lf = %.2lf", num1, num2, result);
40            }
41            break;
42
43        default:
44            printf("Error: Invalid operator");
45    }
46 }

```

**26.** To find the sum of the digits of a given number using while statement. . .

Read  $num$

Initialize  $sum$  to 0.

Repeat until  $num = 0$

    Obtain last digit  $d = num \% 10$

    Add  $d$  to  $sum$

$num = num / 10$

Print  $sum$

C:\Users\Dell\Documents\class work-26.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug [\*]structure.c LCM and HCF.c class work-1.c class work-16.cpp class work-17.c class work-20.c class work-21.c class work-25.c class work-26.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int num, sum = 0, digit;
6
7     printf("Enter a number: ");
8     scanf("%d", &num);
9
10    while (num != 0)
11    {
12        digit = num % 10;
13        sum += digit;
14        num /= 10;
15    }
16
17    printf("Sum of digits: %d", sum);
18
19    return 0;
20
21 }
```

C:\Users\Dell\Documents\class work-26.exe

Enter a number: 5  
Sum of digits: 5  
-----  
Process exited after 8.152 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results

Line: 21 Col: 1 Sel: 0 Lines: 21 Length: 297 Insert Done parsing in 0.031 seconds

