

Print_Stars:

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for `print_stars.c`. The code prints a triangle of stars based on user input. The code editor has syntax highlighting for C/C++ and shows line numbers from 1 to 19. Below the code editor is a log window titled "Logs & others" which is currently empty. The status bar at the bottom shows the file path `C:\C_Programming\Day_2\print_stars.c`, the build configuration `C/C++`, and the current date and time `5/13/2024 10:46 AM`.

```
#include<stdio.h>
#include C:\C_Programming\Day_2\print_stars.c

void main()
{
    int i,j,ns=0;
    system("cls");
    printf("\nEnter the number rows for Stars : ");
    scanf("%d", &ns);
    for(i=1;i<=ns;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
}
```

The screenshot shows the terminal window of the Code::Blocks IDE displaying the execution of the `print_stars` program. The user enters the number of rows as 5. The program then prints a 5x5 triangle of stars. The terminal window also shows the process ID, execution time, and a prompt for the user to press any key to continue.

```
Enter the number rows for Stars : 5
*
*
*
*
*
Process returned 5 (0x5)   execution time : 2.022 s
Press any key to continue.
```

Matrix :

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for `matrix.c`. The code defines a 2x2 matrix and prints its values. The code includes #include directives for `<stdio.h>`, `<conio.h>`, and `<math.h>`. It uses nested loops to input matrix values and print them. The code editor has syntax highlighting for C/C++.

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

void main()
{
    system("cls");
    int i, j, n[2][2];
    printf("Maximum values [2][2]\n");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            scanf("%d", &n[i][j]);
        }
    }
    printf("\nThe Matrix is : \n");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("%d ", n[i][j]);
        }
        printf("\n");
    }
}
```

The logs & others panel shows the build process and messages. The taskbar at the bottom shows the weather (Partly sunny, 87°F) and the date/time (5/13/2024, 10:45 AM).

The screenshot shows a terminal window displaying the output of the `matrix.c` program. The program asks for maximum values of a 2x2 matrix, which are 1, 2, 3, and 4. It then prints the matrix as:

```
Maximum values [2][2]
1
2
3
4

The Matrix is :
1 2
3 4
```

The terminal also shows the process return value (10), execution time (3.787 s), and a prompt to press any key to continue. The taskbar at the bottom shows the weather (Partly sunny, 87°F) and the date/time (5/13/2024, 10:46 AM).

Matrix Multiplication:

The screenshot shows the Code::Blocks IDE interface with the file `matrix_mul.c` open. The code prints two 2x2 matrices and their product. The matrices are defined as 2x2 arrays `a` and `b`, and the result is stored in `c`. The code uses nested loops for input and output.

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

void main()
{
    system("cls");
    int i, j, r, a[2][2], b[2][2], c[2][2];
    printf("\nMatrix Multiplication [2][2]\n");
    printf("First Matrix ::\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%d", &a[i][j]);
        }
        printf("\n");
    }
    printf("Second Matrix ::\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%d", &b[i][j]);
        }
        printf("\n");
    }
    printf("The Matrix a[][] and b[][] are :: \n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }
    printf("\nThe Multiplication is : \n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            c[i][j]=0;
            for(int k=0;k<2;k++)
            {
                c[i][j] = c[i][j] + (a[i][k] * b[k][j]);
            }
        }
    }
}
```

Logs & others

C:\C_Programming\Day_2\matrix_mul.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 41, Col 24, Pos 825 Insert Read/Write default 10:45 AM 5/13/2024

The screenshot shows the Code::Blocks IDE interface with the file `matrix_mul.c` open. The code now includes the logic for multiplying the two matrices `a` and `b` to produce the result matrix `c`. The multiplication is performed using nested loops, where each element `c[i][j]` is calculated as the sum of products of elements from row `i` of `a` and column `j` of `b`.

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

void main()
{
    printf("The Matrix a[][] and b[][] are :: \n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }
    printf("\nThe Multiplication is : \n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            c[i][j]=0;
            for(int k=0;k<2;k++)
            {
                c[i][j] = c[i][j] + (a[i][k] * b[k][j]);
            }
        }
    }
}
```

Logs & others

C:\C_Programming\Day_2\matrix_mul.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 41, Col 24, Pos 825 Insert Read/Write default 10:45 AM 5/13/2024

matrix_mul.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : void

Management Start here print_stars.c matrix.c matrix_mul.c

Projects Files Workspace

```
34     }
35     printf("\n");
36   }
37   for(i=0;i<2;i++)
38   {
39     for(j=0;j<2;j++)
40     {
41       printf("%d ",b[i][j]);
42     }
43     printf("\n");
44   }

45   printf("\nThe Multiplication is : \n");
46   for(i=0;i<2;i++)
47   {
48     for(j=0;j<2;j++)
49     {
50       c[i][j]=0;
51       for(int k=0;k<2;k++)
52       {
53         c[i][j] = c[i][j] + (a[i][k] * b[k][j]);
54       }
55       printf("%d\t", c[i][j]);
56     }
57     printf("\n");
58   }
59 }
60 }
```

Logs & others

Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger Doxygen

C:\C_Programming\Day_2\matrix_mul.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 41, Col 24, Pos 825 Insert Read/Write default

87°F Partly sunny Search ENG IN 10:45 AM 5/13/2024

C:\C_Programming\Day_2\ma

```
Matrix Multiplication [2][2]
First Matrix ::

1
2

3
4

Second Matrix ::

5
6

7
8

The Matrix a[][] and be[][] are ::

1 2
3 4
5 6
7 8

The Multiplication is :

19      22
43      50

Process returned 10 (0xA) execution time : 5.446 s
Press any key to continue.
```

87°F Partly sunny Search ENG IN 10:45 AM 5/13/2024

Factorial :

The screenshot shows the Code::Blocks IDE interface. The main window displays the C code for calculating Factorial. The code includes a main function that reads a number from the user and calls a factorial function. The factorial function calculates the factorial using a for loop and prints each step of the calculation. The logs & others panel shows a build log message indicating "no target" and "no project". The taskbar at the bottom shows various system icons and the date/time.

```
factorial.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here <global> facto(int n) : void
1 #include<stdio.h>
2 int main()
3 {
4     int n;
5     printf("Enter a number : ");
6     scanf("%d", &n);
7     facto (n);
8 }
9 void facto (int n)
10 {
11     int fact = 1;
12     for(int i=1;i<=n;i++)
13     {
14         fact = fact*i;
15         if(i==1)
16             printf("%d ",i);
17         else
18             printf(" * %d ",i);
19     }
20     printf("\nFactorial of is:: %d", fact);
21 }
22
```

Logs & others

Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran info Closed files list Thread

File L... Message

== Build file: "no target" in "no project" (compiler: unknown)

C:\C_Programming\Day_2\factorial.c | C/C++ Windows (CR+LF) WINDOWS-1252 Line 10, Col 2, Pos 141 Insert Read/Write default

95° Partly sunny 1:27 PM 5/13/2024

The screenshot shows a terminal window displaying the execution of the factorial program. The user enters the number 5, and the program calculates the factorial (1*2*3*4*5) and prints the result (120). The terminal also shows the process return status and execution time. The taskbar at the bottom shows various system icons and the date/time.

```
C:\C_Programming\Day_2\fac >
Enter a number : 5
1 * 2 * 3 * 4 * 5
Factorial of is:: 120
Process returned 0 (0x0) execution time : 3.490 s
Press any key to continue.
```

95° Partly sunny 1:25 PM 5/13/2024

Fibonacci Series:

The screenshot shows the Code::Blocks IDE interface. The main window displays a C program for generating a Fibonacci series. The code includes a header file inclusion, a main function that prompts for user input, and a loop that calculates and prints the series. The build log window shows warnings about implicit declarations of 'strlen' and 'incompatible implicit declaration of built-in functions'. The system tray at the bottom indicates it's 94°F and shows other system icons.

```
#include<stdio.h>
void main()
{
    int n, i, a=0, b=1, c;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci series: ");
    for(i=0;i<=n;++i)
    {
        printf("%d ", a);
        c=a+b;
        a=b;
        b=c;
    }
}
```

Logs & others

File L... Message
C:\C_Pro... === Build file: "no target" in "no project" (compiler: unknown)
C:\C_Pro... 12 warning: implicit declaration of function 'strlen' [-Wimpli...
C:\C_Pro... 12 warning: incompatible implicit declaration of built-in func...

C:\C_Programming\Day_2\fib_series.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 8, Col 16, Pos 178 Insert Read/Write default ENG IN 11:26 AM 5/13/2024

The screenshot shows a terminal window within the Code::Blocks IDE displaying the output of the Fibonacci program. It prompts for the number of terms (5), shows the resulting Fibonacci series (0 1 1 2 3 5), and provides execution statistics. The terminal window has a dark theme. The system tray at the bottom indicates it's 94°F and shows other system icons.

```
Enter the number of terms: 5
Fibonacci series: 0 1 1 2 3 5
Process returned 5 (0x5)   execution time : 1.826 s
Press any key to continue.
```

C:\C_Programming 94°F Hot weather Search 11:26 AM 5/13/2024

while loop:

The screenshot shows the Code::Blocks IDE interface. The main window displays a C source code file named 'while.c' with the following content:

```
#include<stdio.h>
void main()
{
    char username[50], domain[50], i;
    again:
    printf("Enter your Username : ");
    scanf("%s", username);
    printf("Enter your domain : ");
    scanf("%s", domain);
    printf("Do you want to continue ? (1 for yes, 0 for No):");
    scanf("%d", &i);
    while(i==1)
    {
        goto again;
    }
}
```

The code uses a goto statement to implement a simple while loop. The logs window shows the build process:

```
File L... Message
==== Build file: "no target" in "no project" (compiler: unknown)
==== Build finished: 0 error(s), 0 warning(s) (0 minute(s), ...)
```

The taskbar at the bottom shows the current weather (90°F, Mostly sunny) and the date/time (12:30 PM, 5/13/2024).

The screenshot shows the Code::Blocks IDE interface with the same 'while.c' file open. The terminal window displays the execution output:

```
Enter your Username : Logesh
Enter your domain : Linux
Do you want to continue ? (1 for yes, 0 for No):1
Enter your Username : Harish
Enter your domain : Analyst
Do you want to continue ? (1 for yes, 0 for No):0

Process returned 0 (0x0)   execution time : 28.165 s
Press any key to continue.
```

The terminal window has a dark background. The taskbar at the bottom shows the current weather (90°F, Mostly sunny) and the date/time (12:30 PM, 5/13/2024).

Do While loop:

The screenshot shows the Code::Blocks IDE interface. The main window displays a C program named 'do_while.c' with the following code:

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

void main()
{
    char username[50], domain[50];
    bool continueInput = true;
    do
    {
        printf("Enter your Username : ");
        scanf("%s", username);
        printf("Enter your domain : ");
        scanf("%s", domain);
        printf("Do you want to continue ? (1 for yes, 0 for No):");
        scanf("%d", &continueInput);
    }
    while (continueInput);
    printf("Thank you\n");
}
```

The code uses standard input/output functions and a do-while loop to repeatedly ask the user for their username and domain until they choose to stop. The logs & others panel shows build messages indicating no errors or warnings.

The screenshot shows a Windows command-line interface window titled 'C:\C_Programming\Day_2\do'. The window displays the output of the 'do_while.c' program:

```
Enter your Username : Logesh
Enter your domain : Linux
Do you want to continue ? (1 for yes, 0 for No):1
Enter your Username : Kavin
Enter your domain : Datacentre
Do you want to continue ? (1 for yes, 0 for No):0
Thank you
```

Below the output, the terminal shows the process return status and a prompt to press any key to continue. The taskbar at the bottom of the screen shows various system icons and the current date and time (5/13/2024, 12:22 PM).

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** dowhile.c - Code::Blocks 20.03
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, Doxygen, Settings, Help
- Project List:** Start here, dup4.c, dowhile.c
- Code Editor:** The code for dowhile.c is displayed:

```
1 #include<stdio.h>
2 #include<stdbool.h>
3
4 void main()
5 {
6     char username[50], domain[50];
7     bool continueInput = true;
8     do
9     {
10         printf("Enter your Username : ");
11         scanf("%s", username);
12         printf("Enter your domain : ");
13         scanf("%s", domain);
14         printf("Do you want to continue ? (1 for yes, 0 for No):");
15         scanf("%d", &continueInput);
16     }
17
18     while (continueInput);
19     printf("Thank you\n");
20 }
21
22
```
- Status Bar:** D:\c\program\test\dowhile.c, C/C++, Windows (CR+LF), WINDOWS-1252, Line 13, Col 29, Pos 284, Insert, Read/Write, default, USA
- Bottom Bar:** Apply changes >
- Taskbar:** Shows various application icons including File Explorer, Task Manager, and several browser and utility icons.
- System Tray:** Shows the date and time (21:54 13-05-2024) and some system icons.

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** dowhile.c - Code::Blocks 20.03
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, Doxygen, Settings, Help
- Toolbar:** Standard Windows-style toolbar with icons for file operations.
- Code Editor:** The main window displays the C code for a do-while loop. The code includes #include directives for stdio.h and stdbool.h, a main function, and a do-while loop that prompts the user for a username and domain, then continues the loop based on user input.

```
1 #include<stdio.h>
2 #include<stdbool.h>
3
4 void main()
5 {
6     char ch;
7     bool bo;
8     do
9     {
10         ch = 'D:\c program\test\dowhile.exe'
11         bo = Enter your Username : sushma
12         do Enter your domain : linux
13         Do you want to continue ? (1 for yes, 0 for No):1
14         Enter your Username : ramya
15         Enter your domain : data science
16         Do you want to continue ? (1 for yes, 0 for No):Enter your Username : Enter your domain : 0
17         Do you want to continue ? (1 for yes, 0 for No):
18     }
19     while(bo);
20     printf("Program ended");
21 }
22
```

- Output Window:** Below the code editor, the output window shows the execution of the program, displaying the user inputs and the program's responses.
- Status Bar:** Shows the file path D:\c program\test\dowhile.c and the date/time 13-05-2024 21:54.
- Bottom Bar:** Contains a green "Apply changes" button and standard Windows taskbar icons.

print all the elements pf a[20] with their Index values:

The screenshot shows the Code::Blocks IDE interface. The top menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The title bar says "printallelements.c - Code::Blocks 20.03". The main window displays the C source code for printing array elements. Below the code editor is a "Logs & others" tab bar with various logs like Ccc, Build log, and Build messages. The status bar at the bottom shows the file path "C:\C_Programming\Day_2\printallelements.c", the current file type "C/C++", encoding "Windows (CR+LF) / WINDOWS-1252", line 1, col 1, pos 0, and other system information.

```
#include<stdio.h>
void main()
{
    int a[20],i;
    printf("Enter 20 array elements:\n");
    for(i=0;i<20;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("Entered array::\n");
    for(i=0;i<20;i++)
    {
        printf("a[%d]=%d",i,a[i]);
        printf("\n");
    }
}
```

The screenshot shows the terminal window of the Code::Blocks IDE. It displays the output of the program, which prints the index and value of each element in the array. The output starts with "Entered array::" followed by a series of assignments from a[0]=1 to a[19]=20. At the end, it shows the process return code and a prompt to press any key to continue. The status bar at the bottom shows the file path "C:\C_Program", the current file type "C/C++", encoding "Windows (CR+LF) / WINDOWS-1252", line 1, col 1, pos 0, and other system information.

```
16
17
18
19
20
Entered array::
a[0]=1
a[1]=2
a[2]=3
a[3]=4
a[4]=5
a[5]=6
a[6]=7
a[7]=8
a[8]=9
a[9]=10
a[10]=11
a[11]=12
a[12]=13
a[13]=14
a[14]=15
a[15]=16
a[16]=17
a[17]=18
a[18]=19
a[19]=20

Process returned 10 (0xA)   execution time : 17.692 s
Press any key to continue.
```

Delete a particular element from an array, a[15] :

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for `delete_array20.c`. The code prompts the user to enter 20 array elements, checks if the entered value is greater than or equal to 20, and if so, prints a message stating deletion is not possible. If not, it performs a deletion operation by shifting all elements after the specified index to the left. Finally, it prints the modified array. The logs panel at the bottom shows various build-related messages.

```
#include<stdio.h>
void main()
{
    int a[20], i, d=15;
    printf("Enter 20 array elements:\n");
    for(i=0; i<20; i++)
    {
        scanf("%d", &a[i]);
    }
    if(d>=20)
    {
        printf("deletion not possible");
    }
    else
    {
        for(i=d; i<20; i++)
        {
            a[i]=a[i+1];
        }
    }
    printf("Entered array::\n");
    for(i=0; i<20; i++)
    {
        printf("a[%d]=%d", i, a[i]);
        printf("\n");
    }
}
```

The screenshot shows a terminal window displaying the output of the `delete_array20.c` program. The user enters 20 array elements. The program then prints the entered array, followed by the modified array where the element at index 15 has been deleted. The terminal also shows the process return status and a prompt for the user to press any key to continue.

```
16
17
18
19
20
Entered array::
a[0]=1
a[1]=2
a[2]=3
a[3]=4
a[4]=5
a[5]=6
a[6]=7
a[7]=8
a[8]=9
a[9]=10
a[10]=11
a[11]=12
a[12]=13
a[13]=14
a[14]=15
a[15]=17
a[16]=18
a[17]=19
a[18]=20
a[19]=11015120

Process returned 10 (0xA)   execution time : 16.100 s
Press any key to continue.
```

Find if there are any duplicates in a[20]:

The screenshot shows the Code::Blocks IDE interface. The main window displays the source code for 'duplicate.c'. The code reads 20 integers from the user, stores them in an array, and then compares each element with every other element in the array to find duplicates. The code uses nested loops and an if-statement to check for equality. The 'Logs & others' panel at the bottom shows build logs indicating no errors or warnings.

```
#include<stdio.h>
void main()
{
    int a[20],i,j;
    printf("Enter 20 array elements:\n");
    for(i=0;i<20;i++)
    {
        scanf("%d",&a[i]);
    }
    printf("duplicate elements in the array::\n");
    for(i=0;i<20;i++)
    {
        for(j=i+1;j<20;j++)
        {
            if (a[i]==a[j])
            {
                printf("%d ",a[i]);
                break;
            }
        }
    }
}
```

Logs & others

File L... Message
==== Build file: "no target" in "no project" (compiler: unknown)
==== Build finished: 0 error(s), 0 warning(s) (0 minute(s), ...)

The screenshot shows a terminal window running under the Windows operating system. It displays the output of the 'duplicate.c' program. The user is prompted to enter 20 array elements. After entering the numbers, the program outputs "duplicate elements in the array:" followed by the number 11, which is a duplicate in the input sequence. The terminal also shows the execution time and a prompt to press any key to continue.

```
Enter 20 array elements:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
11
15
16
17
18
19
duplicate elements in the array::
11
Process returned 20 (0x14)   execution time : 17.948 s
Press any key to continue.
```

search [10] from a[20]:

The screenshot shows the Code::Blocks IDE interface. The main window displays the code for `search.c`:#include<stdio.h>
void main()
{
 int a[20],i;
 printf("Enter 20 array elements:\n");
 for(i=0;i<20;i++)
 {
 scanf("%d",&a[i]);
 }
 printf("10th element is::\n");
 for(i=0;i<20;i++)
 {
 if(i==10)
 {
 printf("%d",a[i]);
 }
 }
}The code prompts the user to enter 20 array elements and then prints the 10th element. Below the code editor is the "Logs & others" panel, which shows the build log:File L... Message
==== Build file: "no target" in "no project" (compiler: unknown)
==== Build finished: 0 error(s), 0 warning(s) (0 minute(s), ...)The status bar at the bottom indicates the current weather and date.

The screenshot shows the terminal window of the Code::Blocks IDE displaying the output of the `search.c` program:C:\C_Programming\Day_2>search.c
Enter 20 array elements:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
10th element is::
11
Process returned 2 (0x2) execution time : 16.897 s
Press any key to continue.
|The terminal also shows the system status bar with weather information and a taskbar with various application icons.

