

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

- Project Bar:** Shows tabs for "copy string.cpp" and "binary search.cpp".
- Code Editor:** Displays the "binary search.cpp" file. The code implements a binary search algorithm. It includes comments explaining the search logic: checking if the target value is at the middle index, ignoring the left half if the target is greater than the mid-value, and ignoring the right half if the target is smaller. It handles the case where the element is not found by returning -1. The driver code initializes an array with values {2, 3, 4, 10, 40}, sets the search range from index 0 to n-1, and searches for the value 10. The output is printed using a ternary operator based on the result.
- Output Window:** Shows the terminal output of the compiled program. It displays the message "Element is present at index 3", followed by a separator line, and the program exit information: "Process exited after 0.04918 seconds with return value 0". It also prompts the user to press any key to continue.
- Compiler Log:** Located at the bottom, it shows the compilation results:
  - Errors: 0
  - Warnings: 0
  - Output Filename: C:\Users\kondu\OneDrive\Documents\binary search.exe
  - Output Size: 128.470703125 KiB
  - Compilation Time: 0.23s

AutoSave Off Document1 - Word Search monisha.konduru MK

C:\Users\kondu\OneDrive\Documents\reverse.cpp - [Executing] - Dev C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.5.2 64-bit Release

(globals)

Project Classes Debug mex end min.cpp Untitled2.cpp Untitled3 Untitled4.cpp Untitled5.cpp Untitled11.cpp Untitled12.cpp Untitled13.cpp Untitled14

```
1 // C program to reverse the string in C using loops
2 #include <stdio.h>
3 #include <string.h>
4
5 int main()
6 {
7     // string to be reversed.
8     char str[100] = "string";
9
10    printf("Original String: %s\n", str);
11
12    // string Length
13    int len = strlen(str);
14
15    // for Loop
16    for (int i = 0, j = len - 1; i <= j; i++, j--) {
17        // swapping characters
18    }
}
```

Comments Editing Share

Original String: string  
Reversed String: gnirts

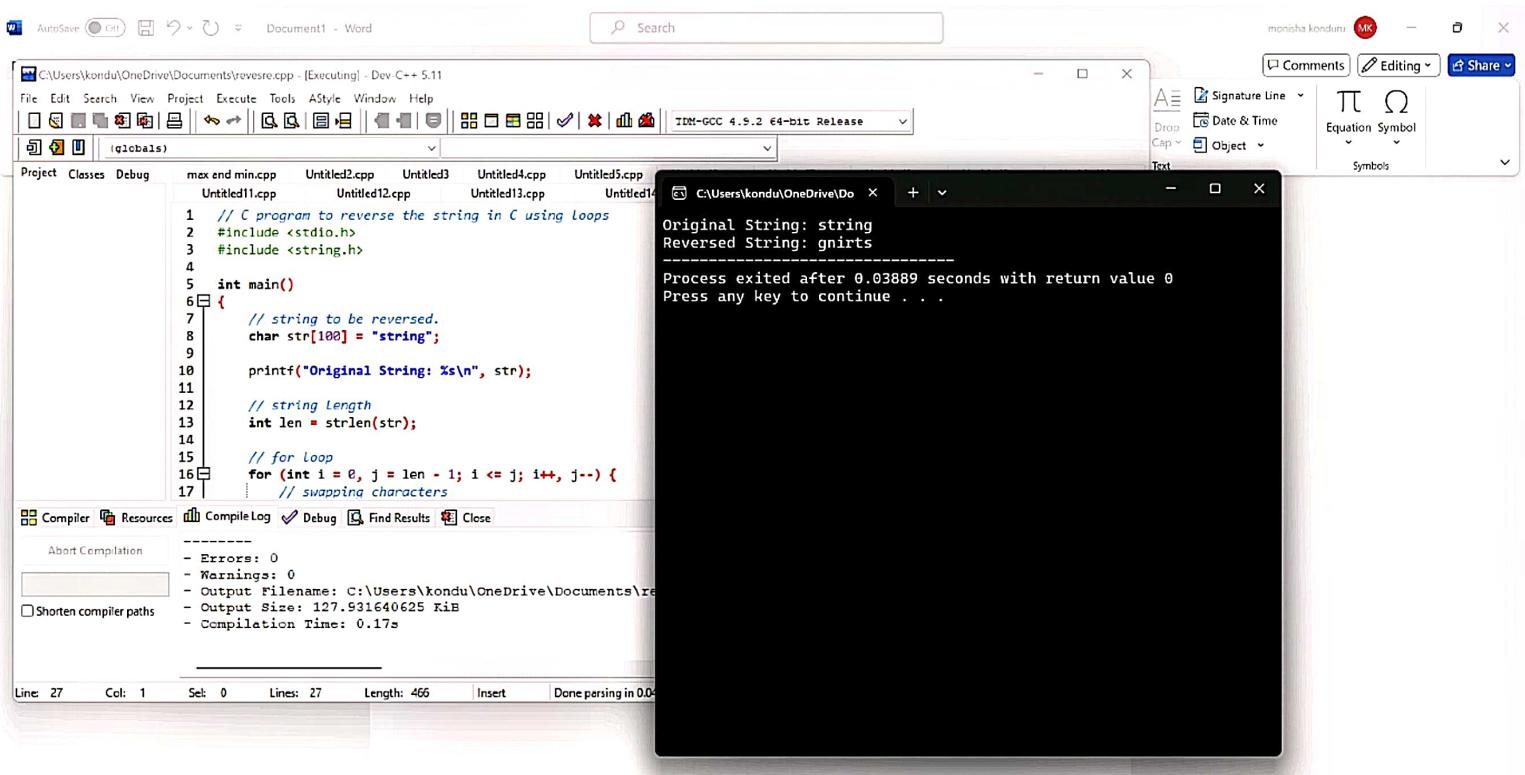
Process exited after 0.03889 seconds with return value 0  
Press any key to continue . . .

Compiler Resources CompileLog Debug Find Results Close

Abort Compilation

Shorten compiler paths

Line: 27 Col: 1 Sel: 0 Lines: 27 Length: 466 Insert Done parsing in 0.0s



The screenshot shows the Code::Blocks IDE interface. The top menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help, and a language-specific menu like TDM-GCC. Below the menu is a toolbar with various icons for file operations. The main window has tabs for Project, Classes, Debug, and copy string.cpp (which is currently selected). The code editor displays the following C program:

```
copy string.cpp binary search.cpp reverse string.cpp length of string.cpp
1 // C program to find the length of string
2 #include <stdio.h>
3 #include <string.h>
4
5 int main()
6 {
7     char Str[1000];
8     int i;
9
10    printf("Enter the String: ");
11    scanf("%s", Str);
12
13    for (i = 0; Str[i] != '\0'; ++i);
14
15    printf("Length of Str is %d", i);
16
17    return 0;
18 }
```

```
C:\Users\kondu\OneDrive\Do + v  
Enter the String: moni  
Length of Str is 4  
-----  
Process exited after 2.486 seconds with return value 0  
Press any key to continue . . . |
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation	<hr/> <ul style="list-style-type: none"><li>- Errors: 0</li><li>- Warnings: 0</li><li>- Output Filename: C:\Use</li><li>- Output Size: 128.122070</li><li>- Compilation Time: 0.16s</li></ul>
<input type="checkbox"/> Shorten compiler paths	

C:\Users\kondu\OneDrive\Documents\strassens multiplication.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

copy string.cpp binary search.cpp reverse string.cpp length of string.cpp strassens multiplication.cpp

```

23     printf("%d\t", a[i][j]);
24 }
25
26 printf("\nThe second matrix is\n");
27 for(i = 0; i < 2; i++){
28     printf("\n");
29     for(j = 0; j < 2; j++)
30         printf("%d\t", b[i][j]);
31 }
32
33 m1= (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
34 m2= (a[1][0] + a[1][1]) * b[0][0];
35 m3= a[0][0] * (b[0][1] - b[1][1]);
36 m4= a[1][1] * (b[1][0] - b[0][0]);
37 m5= (a[0][0] + a[0][1]) * b[1][1];
38 m6= (a[1][0] - a[0][0]) * (b[0][0]+b[0][1]);
39 m7= (a[0][1] - a[1][1]) * (b[1][0]+b[1][1]);
40
41 c[0][0] = m1 + m4- m5 + m7;
42 c[0][1] = m3 + m5;
43 c[1][0] = m2 + m4;
44 c[1][1] = m1 - m2 + m3 + m6;
45
46 printf("\nAfter multiplication using Strassen's algorithm \n");
47 for(i = 0; i < 2 ; i++){
48     printf("\n");
49     for(j = 0;j < 2; j++)
50         printf("%d\t", c[i][j]);
51 }
52
53 return 0;
54

```

The first matrix is

1	2
3	4

The second matrix is

5	6
7	8

After multiplication using Strassen's algorithm

19	22
43	50

Process exited after 10.45 seconds with return value 0  
Press any key to continue . . . |

Compiler Resources Compiler Log Debug Find Results Close

Abort Compilation

Shorten compiler paths

-----  
- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\kondu\OneDrive\Documents\strassens multiplication.exe  
- Output Size: 129.970703125 Kib  
- Compilation Time: 0.17s

The screenshot shows a Windows terminal window titled 'C:\Users\kondu\OneDrive\Do' with the following output:

```
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13

Process exited after 0.05982 seconds with return value 0
Press any key to continue . . .
```

Below the terminal window, the code editor displays the C++ file 'merge sort.cpp' with line numbers 74 to 95. The code implements a merge sort algorithm with utility functions for printing arrays and a driver main function.

```
74     mergeSort(arr, l, m);
75     mergeSort(arr, m + 1, r);
76
77     merge(arr, l, m, r);
78 }
79
80 // UTILITY FUNCTIONS
81 // Function to print an array
82 void printArray(int A[], int size)
83 {
84     int i;
85     for (i = 0; i < size; i++)
86         printf("%d ", A[i]);
87     printf("\n");
88 }
89
90 // Driver code
91 int main()
92 {
93     int arr[] = { 12, 11, 13, 5, 6, 7 };
94     int arr_size = sizeof(arr) / sizeof(arr[0]);
95
96     printf("Given array is \n");
97     printArray(arr, arr_size);
98
99     mergeSort(arr, 0, arr_size - 1);
100
101    printf("\nSorted array is \n");
102    printArray(arr, arr_size);
103    return 0;
104 }
```

Compile Log Debug Find Results Close

```
Errors: 0
Warnings: 0
Output Filename: C:\Users\kondu\OneDrive\Documents\merge sort.exe
Output Size: 130.373046875 KiB
Compilation Time: 0.20s
```

max and min.cpp - [Executing] - Dev-C++ 5.11

File Execute Tools AStyle Window Help

gcd1.cpp gcd1.cpp reverse.cpp

max and min.cpp Untitled2.cpp Untitled3.cpp Untitled4.cpp Untitled5.cpp Untitled6.cpp Untitled7.cpp Untitled8.cpp Untitled9.cpp Untitled10.cpp Untitled11.cpp Untitled12.cpp Untitled13.cpp Untitled14.cpp gcd.cpp

```
#include <stdio.h>

// Function to find minimum and maximum elements in an array
void findMinMax(int arr[], int n) {
    int min = arr[0];
    int max = arr[0];

    // Traverse array elements
    for (int i = 1; i < n; i++) {
        if (arr[i] < min)
            min = arr[i];
        else if (arr[i] > max)
            max = arr[i];
    }

    // Print minimum and maximum elements
    printf("min=%d max=%d\n", min, max);
}

int main()
{
    // Input array
    int arr[] = { 1, 423, 6, 46, 34, 23, 13, 53, 4 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // Find minimum and maximum elements
    findMinMax(arr, n);

    return 0;
}
```

min=1 max=423

Process exited after 0.06095 seconds with return value 0  
Press any key to continue . . .

Compile Log Debug Find Results Close

-----

Errors: 0  
Warnings: 0  
Output: C:\Users\kondu\OneDrive\Documents\max and min.exe

\Users\kondu\OneDrive\Documents\knapsack greedy.cpp - [Executing] - Dev-C++ 5.11

Edit Search View Project Execute Tools AStyle Window Help

File View Insert Edit Project Properties Run Build Options Compiler Resources

copy string.cpp binary search.cpp reverse string.cpp length of string.cpp stressens multiplication.cpp merge sort.cpp max and min.cpp generate prime.cpp knapsack greedy.cpp

```
20 for (j = i + 1; j < n; j++)
21     if (ratio[i] < ratio[j])
22     {
23         temp = ratio[j];
24         ratio[j] = ratio[i];
25         ratio[i] = temp;
26
27         temp = weight[j];
28         weight[j] = weight[i];
29         weight[i] = temp;
30
31         temp = profit[j];
32         profit[j] = profit[i];
33         profit[i] = temp;
34     }
35
36     printf("Knapsack problems using Greedy Algorithm:\n");
37     for (i = 0; i < n; i++)
38     {
39         if (weight[i] > capacity)
40             break;
41         else
42         {
43             Totalvalue = Totalvalue + profit[i];
44             capacity = capacity - weight[i];
45         }
46     }
47     if (i < n)
48         Totalvalue = Totalvalue + (ratio[i]*capacity);
49     printf("\nThe maximum value is :%f\n",Totalvalue);
50 }
51 
```

C:\Users\kondu\OneDrive\Do

Enter the number of items :2  
Enter Weight and Profit for item[0] :  
50 23  
Enter Weight and Profit for item[1] :  
45 29  
Enter the capacity of knapsack :  
100  
Knapsack problems using Greedy Algorithm:  
The maximum value is :52.000000

-----  
Process exited after 18.48 seconds with return value 0  
Press any key to continue . . .

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

- Project Bar:** Shows multiple projects: copy string.cpp, binary search.cpp, reverse string.cpp, length of string.cpp, stressens multiplication.cpp, merge sort.cpp, max and min.cpp, and generate prime.cpp.
- Code Editor:** Displays the source code for `generate prime.cpp`. The code implements a prime number generator. It includes a helper function `isPrime` and a driver function `main` that prints prime numbers from 1 to N (50 in this case). The output window shows the generated prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.
- Compiler Log:** Shows the compilation results:
  - Errors: 0
  - Warnings: 0
  - Output Filename: C:\Users\kondu\OneDrive\Documents\generate prime.exe
  - Output Size: 120.4609375 Kib
  - Compilation Time: 0.20s

File Edit Search View Project Execute Tools AStyle Window Help

IIM-GCC 4.9.2 64-bit Release

(qlcbals)

Project Classes Debug MST using greedy technique.cpp

```
110 |   / \ /  
111 | 2-----3  
112 | | 4 * /  
113 int V = 4; // Number of vertices in graph  
114 int E = 5; // Number of edges in graph  
115 struct Graph* graph = createGraph(V, E);  
116  
117 // add edge 0-1  
118 graph->edge[0].src = 0;  
graph->edge[0].dest = 1;  
graph->edge[0].weight = 10;  
119  
120 // add edge 0-2  
121 graph->edge[1].src = 0;  
graph->edge[1].dest = 2;  
graph->edge[1].weight = 6;  
122  
123 // add edge 0-3  
124 graph->edge[2].src = 0;  
graph->edge[2].dest = 3;  
graph->edge[2].weight = 5;  
125  
126 // add edge 1-3  
127 graph->edge[3].src = 1;  
graph->edge[3].dest = 3;  
graph->edge[3].weight = 15;  
128  
129 // add edge 2-3  
130 graph->edge[4].src = 2;  
graph->edge[4].dest = 3;  
graph->edge[4].weight = 4;  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141
```

C:\Users\kondu\OneDrive\Do

Following are the edges in the constructed MST  
2 -- 3 == 4  
0 -- 3 == 5  
0 -- 1 == 10  
-----  
Process exited after 0.04616 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compiler Log Debug Find Results Close

About Compilation

- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\kondu\OneDrive\Documents\MST using greedy technique.exe  
- Output Size: 130.4794921075 KiB  
- Compilation Time: 1.69s

C:\Users\kondu\OneDrive\Documents\BST.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools A Style Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug BST.cpp

```
18 for (int i = 1; i <= n-len+1; i++) {
19     int j = i + len - 1;
20     C[i][j] = FLT_MAX; // Initialize as maximum float value
21     float sum_p = 0;
22     for (int k = i; k <= j; k++) {
23         sum_p += probabilities[k-1];
24     }
25     for (int r = i; r <= j; r++) {
26         float cost = C[i][r-1] + C[r+1][j] + sum_p;
27         if (cost < C[i][j]) {
28             C[i][j] = cost;
29         }
30     }
31 }
32
33 // Return the optimal BST cost
34 return C[1][n];
35
36
37
38 int main() {
39     // Example keys and probabilities
40     float keys[] = {10, 12, 20};
41     float probabilities[] = {0.1, 0.2, 0.3};
42     int n = sizeof(keys) / sizeof(keys[0]);
43
44     // Compute optimal BST cost
45     float result = optimalBST(keys, probabilities, n);
46
47     // Print the result
48     cout << "Optimal BST cost: " << result << endl;
49
50
51 }
```

C:\Users\kondu\OneDrive\Do

Optimal BST cost: 1

-----

Process exited after 0.05632 seconds with return value 0

Press any key to continue . . . |

C:\Users\kondu\OneDrive\Documents\reversednumber.cpp - [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 BST.cpp [1] dynamic programming.cpp reversednumber.cpp

(global)

```
1 #include <stdio.h>
2
3 int reverseNumber(int num) {
4     int reversedNum = 0;
5
6     // Iterate until num becomes 0
7     while (num != 0) {
8         // Get the last digit of num
9         int remainder = num % 10;
10
11         // Append this digit to reversedNum
12         reversedNum = reversedNum * 10 + remainder;
13
14         // Remove the last digit from num
15         num = num / 10;
16     }
17
18     return reversedNum;
19 }
20
21 int main() {
22     int num;
23
24     // Input the number to be reversed
25     printf("Enter a number to reverse: ");
26     scanf("%d", &num);
27
28     // Reverse the number
29     int reversed = reverseNumber(num);
30
31     // Output the reversed number
32     printf("Reversed number: %d\n", reversed);
33
34     return 0;
}
```

C:\Users\kondu\OneDrive\Do x + v

Enter a number to reverse: 5678  
Reversed number: 8765

-----  
Process exited after 3.513 seconds with return value 0  
Press any key to continue . . .

C:\Users\kondu\OneDrive\Documents\dynamic programming.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug BST.cpp dynamic programming.cpp

```
1 #include <stdio.h>
2
3 // Function to calculate binomial coefficient C(n, k) using dynamic programming
4 int binomialCoefficient(int n, int k) {
5     int C[n + 1][k + 1];
6
7     // Calculate binomial coefficients using dynamic programming approach
8     for (int i = 0; i <= n; i++) {
9         for (int j = 0; j <= i && j <= k; j++) {
10            if (j == 0 || j == i)
11                C[i][j] = 1;
12            else
13                C[i][j] = C[i - 1][j - 1] + C[i - 1][j];
14        }
15    }
16
17    return C[n][k];
18 }
19
20 int main() {
21     int n = 5, k = 2; // Example: C(5, 2)
22     int result = binomialCoefficient(n, k);
23     printf("C(%d, %d) = %d\n", n, k, result);
24
25     return 0;
26 }
```

C:\Users\kondu\OneDrive\Do X + v

C(5, 2) = 10

-----

Process exited after 0.05249 seconds with return value 0

Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

About Compilation

Abort Compilation

Shorten compiler paths

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\kondu\OneDrive\Documents\dynamic programming.exe
- Output Size: 120.956046875 Kib
- Compilation Time: 0.19s
```

C:\Users\kondu\OneDrive\Documents\tsp.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug max and min.cpp Untitled2.cpp Untitled3 Untitled4.cpp Untitled5.cpp Untitled12.cpp Untitled13.cpp Untitled14.cpp gcd.cpp tsp.cpp

```
35     // Memoize the result
36     return dp[mask][pos] = min_cost;
37 }
38 }
39
40 int main() {
41     // Input number of cities
42     printf("Enter the number of cities: ");
43     scanf("%d", &n);
44
45     // Input distance matrix
46     printf("Enter the distance matrix (%d x %d):\n", n, n)
47     for (int i = 0; i < n; i++) {
48         for (int j = 0; j < n; j++) {
49             scanf("%d", &dist[i][j]);
50         }
51     }
52
53     // Initialize dp table with -1 (uncomputed)
54     for (int i = 0; i < (1 << n); i++) {
55         for (int j = 0; j < n; j++) {
56             dp[i][j] = -1;
57         }
58     }
59
60     // Start TSP from city 0
61     int min_cost = tsp(1, 0); // Start with mask 1 (only city 0)
62
63     // Output the minimum cost (shortest path)
64     printf("Minimum cost for TSP: %d\n", min_cost);
65 }
```

Enter the number of cities: 4  
Enter the distance matrix (4 x 4):  
10  
15  
20  
10  
0  
35  
25  
15  
35  
0  
30  
20  
25  
30  
0  
Minimum cost for TSP: 80

-----  
Process exited after 47.32 seconds with return value 0  
Press any key to continue . . .

File Edit Search View Project Execute Tools AStyle Window Help

File Edit View Project Tools Help

BST.cpp [dynamic programming.cpp reversednumber.cpp perfect number.cpp]

```
1 #include <stdio.h>
2
3 // Function to check if a number is perfect
4 int isPerfect(int num) {
5     int sum = 0;
6
7     // Find all divisors and sum them
8     for (int i = 1; i < num; i++) {
9         if (num % i == 0) {
10             sum += i;
11         }
12     }
13
14     // Check if sum of divisors equals the number itself
15     return (sum == num);
16 }
17
18 int main() {
19     int limit;
20
21     // Input the upper limit to find perfect numbers
22     printf("Enter the upper limit to find perfect numbers:");
23     scanf("%d", &limit);
24
25     // Iterate through numbers to find perfect numbers
26     printf("Perfect numbers up to %d:\n", limit);
27     for (int i = 1; i <= limit; i++) {
28         if (isPerfect(i)) {
29             printf("%d\n", i);
30         }
31     }
32
33     return 0;
34 }
```

C:\Users\kondu\OneDrive\Do

```
Enter the upper limit to find perfect numbers: 56
Perfect numbers up to 56:
6
28
-----
Process exited after 2.56 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\kondu\OneDrive\Documents\perfect number.exe

C:\Users\kondu\OneDrive\Documents\floydstrinagle.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug floydstrinagle.cpp

```
49 // Mark the picked vertex as processed
50 sptSet[u] = true;
51
52 // Update dist value of the adjacent vertices of the picked vertex.
53 for (int v = 0; v < V; v++) {
54     // Update dist[v] only if it is not in sptSet, there is an edge from
55     // u to v, and total weight of path from src to v through u is
56     // smaller than current value of dist[v]
57     if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX
58         && dist[u] + graph[u][v] < dist[v])
59         dist[v] = dist[u] + graph[u][v];
60 }
61
62
63 // print the constructed distance array
64 printSolution(dist);
65
66 }
67
68 int main() {
69     // Example graph represented using adjacency matrix
70     int graph[V][V] = {
71         {0, 4, 0, 0, 0, 0},
72         {4, 0, 8, 0, 0, 0},
73         {0, 8, 0, 7, 0, 4},
74         {0, 0, 7, 0, 9, 14},
75         {0, 0, 0, 9, 0, 10},
76         {0, 0, 4, 14, 10, 0}
77     };
78
79     dijkstra(graph, 0); // Calculate shortest paths from source vertex 0
80
81     return 0;
82 }
```

Vertex Distance from Source

Vertex	Distance from Source
0	0
1	4
2	12
3	19
4	26
5	16

-----

Process exited after 0.04304 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Options Output Find Results Close

```
Project Classes Debug BST.cpp [+] dynamic programming.cpp | reversednumber.cpp | perfect number.cpp | pattern reverse.cpp
```

```
31     printf("\n");
32
33     // Lower part of the cross (excluding middle line)
34     for (i = 1; i <= n; i++) {
35         // Print leading spaces
36         for (k = n; k > i; k--) {
37             printf(" ");
38         }
39
40         // Print numbers from 1 to i
41         for (j = 1; j <= i; j++) {
42             printf("%d", j);
43             if (j < i) {
44                 printf(" ");
45             }
46         }
47
48         printf("\n");
49     }
50 }
51
52 int main() {
53     int n;
54
55     // Input the value of n
56     printf("Enter the value of n: ");
57     scanf("%d", &n);
58
59     // Print the reversed cross pattern
60     printCrossPatternReverse(n);
61
62     return 0;
63 }
```

```
C:\Users\kondu\OneDrive\Do X + - X
Enter the value of n: 4
1 2 3 4
1 2 3
1 2
1
1 2 3 4
1
1 2
1 2 3
1 2 3 4
-----
Process exited after 2.138 seconds with return value 0
Press any key to continue . . .
```

```
Compiler(1) Resources Compile Log Debug Find Results Close
Line Col File Message
19 C:\Users\kondu\OneDrive\Documents\pattern reverse.cpp [Warning] extra tokens at end of #include directive
```

C:\Users\kondu\OneDrive\Documents\dijkstra garph.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug floyds triagle.cpp calculate factoriel.cpp pascal triagle.cpp dijkstra garph.cpp

```
49     // Mark the picked vertex as processed
50     sptSet[u] = true;
51
52     // Update dist value of the adjacent vertices of the picked vertex.
53     for (int v = 0; v < V; v++) {
54         // Update dist[v] only if it is not in sptSet, there is an edge from
55         // u to v, and total weight of path from src to v through u is
56         // smaller than current value of dist[v]
57         if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX
58             && dist[u] + graph[u][v] < dist[v])
59             dist[v] = dist[u] + graph[u][v];
60     }
61
62     // print the constructed distance array
63     printSolution(dist);
64
65 }
66
67 int main() {
68     // Example graph represented using adjacency matrix
69     int graph[V][V] = {
70         {0, 4, 0, 0, 0, 0},
71         {4, 0, 8, 0, 0, 0},
72         {0, 8, 0, 7, 0, 4},
73         {0, 0, 7, 0, 9, 14},
74         {0, 0, 0, 9, 0, 10},
75         {0, 0, 4, 14, 10, 0}
76     };
77
78     dijkstra(graph, 0); // Calculate shortest paths from source vertex 0
79
80     return 0;
81 }
```

Vertex Distance from Source

Vertex	Distance from Source
0	0
1	4
2	12
3	19
4	26
5	16

-----  
Process exited after 0.04266 seconds with return value 0  
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

C:\Users\kondu\OneDrive\Documents\pascal trainagle.cpp - [Executing] - Dev-C++ 5.11

Edit Search View Project Execute Tools AStyle Window Help

File Project Classes Debug floyds trainagle.cpp calculate factorial.cpp pascal trainagle.cpp

```
9 }
10
11 // Function to calculate binomial coefficient C(n, k)
12 int binomialCoeff(int n, int k) {
13     return factorial(n) / (factorial(k) * factorial(n - k));
14 }
15
16 // Function to print Pascal's Triangle
17 void printPascalTriangle(int numRows) {
18     for (int i = 0; i < numRows; i++) {
19         // Print spaces for alignment
20         for (int space = 1; space <= numRows - i; space++)
21             printf(" ");
22
23         // Print values in a row
24         for (int j = 0; j <= i; j++) {
25             printf("%4d", binomialCoeff(i, j));
26
27         printf("\n");
28     }
29 }
30
31 int main() {
32     int numRows;
33
34     // Input the number of rows for Pascal's Triangle
35     printf("Enter the number of rows for Pascal's Triangle: ");
36     scanf("%d", &numRows);
37
38     // Print Pascal's Triangle with the specified number of rows
39     printPascalTriangle(numRows);
40
41     return 0;
42 }
```

Project Classes Debug floyds trainagle.cpp calculate factorial.cpp pascal trainagle.cpp

File C:\Users\kondu\OneDrive\Do +

Enter the number of rows for Pascal's Triangle: 5

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

-----

Process exited after 1.462 seconds with return value 0

Press any key to continue . . .

Compiler Resources

Abort Compilation

Shorten compiler paths

Compile Log Debug Find Results Close

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\kondu\OneDrive\Documents\pascal trainagle.exe
- Output Size: 129.404296075 KiB
- Compilation Time: 0.17s
```

(kondu\OneDrive\Documents\Untitled6.cpp - [Executing] - Dev-C++ 5.11

Search View Project Execute Tools AStyle Window Help

TIK-GCC 4.9.2 64-bit Release

Globals

lasses Debug max and min.cpp Untitled2.cpp Untitled3 Untitled4.cpp Untitled5.cpp Untitled6.cpp

```
21     minE = arr[i];
22 }
23
24 // If current element is greater
25 // than maxE then update it
26 if (arr[i] > maxE) {
27     maxE = arr[i];
28 }
29
30
31 // Print the minimum and maximum element
32 printf("The minimum element is %d", minE);
33 printf("\n");
34 printf("The maximum element is %d", maxE);
35
36 return;
37 }
38
39 // Driver Code
40 int main()
41 {
42
43 // Given array
44 int arr[] = { 1, 2, 4, -1 };
45
46 // Length of the array
47 int N = sizeof(arr) / sizeof(arr[0]);
48
49 // Function call
50 findMinimumMaximum(arr, N);
51
52 return 0;
53 }
```

C:\Users\kondu\OneDrive\Do +

```
The minimum element is -1
The maximum element is 4
-----
Process exited after 0.0564 seconds with return value 0
Press any key to continue . . .
```

File Resources Compile Log Debug Find Results Close

(kondu\OneDrive\Documents\sum of digits.cpp - [Executing] - Dev-C++ 5.11

Search View Project Execute Tools AStyle Window Help

TIIM-GCC 4.9.2 64-bit Release

Classes Debug floydstrinagle.cpp calculatefactoriel.cpp pascalstrinagle.cpp dijkstra graph.cpp sum of digits.cpp

```
1 #include <stdio.h>
2
3 // Function to calculate the sum of digits of a number
4 int sumOfDigits(int number) {
5     int sum = 0;
6
7     // Loop until number becomes 0
8     while (number != 0) {
9         // Add the last digit to sum
10        sum += number % 10;
11
12        // Remove the last digit from number
13        number /= 10;
14    }
15
16    return sum;
17 }
18
19 int main() {
20     int number;
21
22     // Input the number from the user
23     printf("Enter a number: ");
24     scanf("%d", &number);
25
26     // Calculate the sum of digits
27     int sum = sumOfDigits(number);
28
29     // Print the result
30     printf("Sum of digits of %d is %d\n", number, sum);
31
32     return 0;
33 }
```

Enter a number: 5678  
Sum of digits of 5678 is 26

Process exited after 3.329 seconds with return value 0  
Press any key to continue . . .

File Resources Compile Log Debug Find Results Close

Compilation

compiler paths

```
-----
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\kondu\OneDrive\Documents\sum of digits.exe
- Output Size: 120.6357421875 KiB
- Compilation Time: 0.16s
```

C:\Users\kondu\OneDrive\Documents\sum of subsets.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug insert a number.cpp sum of subsets.cpp

```
10 void generateSubsets(int index, int sum) {
11     if (index == n) {
12         // Base case: Print current subset and its sum
13         printf("{ ");
14         for (int i = 0; i < index; i++) {
15             printf("%d ", subset[i]);
16         }
17         printf("} Sum: %d\n", sum);
18         return;
19     }
20
21     // Include current element in the subset and recurse
22     subset[index] = set[index];
23     generateSubsets(index + 1, sum + set[index]);
24
25     // Exclude current element from the subset and recurse
26     generateSubsets(index + 1, sum);
27 }
28
29 int main() {
30     // Example set initialization
31     printf("Enter number of elements in the set: ");
32     scanf("%d", &n);
33
34     printf("Enter the elements of the set:\n");
35     for (int i = 0; i < n; i++) {
36         scanf("%d", &set[i]);
37     }
38
39     printf("Subsets and their sums:\n");
40     generateSubsets(0, 0); // Start generating subsets from index 0 with initial sum 0
41
42     return 0;
43 }
```

Enter number of elements in the set: 2  
Enter the elements of the set:  
2  
2  
Subsets and their sums:  
{ 2 2 } Sum: 4  
{ 2 2 } Sum: 2  
{ 2 2 } Sum: 2  
{ 2 2 } Sum: 0

-----  
Process exited after 3.346 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

File Edit Search View Project Execute Tools AStyle Window Help

Untitled7.cpp

```
53     board[row][col] = 1;
54
55     // Recur to place rest of the queens
56     if (solveNQueens(row + 1))
57         return true;
58
59     // If placing queen in board[row][col] doesn't lead to a solution,
60     // then backtrack: remove queen from board[row][col]
61     board[row][col] = 0;
62 }
63
64
65 // If no column allows a queen to be placed in this row, return false
66 return false;
67 }

68 int main() {
69     // Initialize the board with zeros
70     for (int i = 0; i < N; i++) {
71         for (int j = 0; j < N; j++) {
72             board[i][j] = 0;
73         }
74     }
75
76
77     // Call the recursive function to solve N Queens problem starting from the first
78     if (solveNQueens(0)) {
79         printf("Solution found:\n");
80         printBoard();
81     } else {
82         printf("No solution exists for N = %d.\n", N);
83     }
84
85     return 0;
86 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Shorten compiler paths

C:\Users\kondu\OneDrive\Do

Solution found:

1	0	0	0	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	1	0	0	0	0
0	0	0	0	0	0	1
0	1	0	0	0	0	0
0	0	0	1	0	0	0

Process exited after 0.07181 seconds with return value 0  
Press any key to continue . . .

-----  
- Errors: 0  
- Warnings: 0  
- Output Filename: C:\Users\kondu\OneDrive\Documents\Untitled7.exe  
- Output Size: 129.8876953125 KiB  
- Compilation Time: 0.22s

File Project Execute Tools AStyle Window Help

insert a number.cpp sum of subsets.cpp graph coloring.cpp container loading.cpp

```
33 // Start loading into a new truck
34 truckCount++;
35 printf("\nTruck %d:\n", truckCount + 1);
36 printf("Container %d (Weight: %d)\n", containers[i].id, containers[i].weight);
37 currentWeight = containers[i].weight; // Reset current weight
38
39 } Enter number of containers: 2
40 Enter details for each container (id, weight, capacity):
41 i Container 1: 2
42 2
43 Container 2: 2
44 2
45 Enter maximum weight capacity of each truck: 2
46
47 Loading containers into trucks...
48
49 Truck 1:
50 Container 2 (Weight: 2)
51
52 Truck 2:
53 Container 1 (Weight: 2)
54
55 Total trucks used: 2
56
57 -----
58 Process exited after 14.12 seconds with return value 0
59 Press any key to continue . . .
60
61
62
63
64
65
66
```

Sources

- Error
- Warning
- Output Filename: C:\Users\kondu\OneDrive\Documents\container loading.exe
- Output Size: 130.544921875 KiB
- Compilation Time: 0.17s

2 Sat 0 Lines: 66 Length: 2432 Insert Done parsing in 0.017 seconds

```
void hamiltonianCircuit() {
    path[0] = 0; // Start from vertex 0 as the first vertex in the path
    for (int i = 1; i < numVertices; i++)
        path[i] = -1; // Initialize all other vertices as not included in the path
}
Enter number of vertices: 2
Enter the adjacency matrix (2 x 2):
2
2
2
2
2
Hamiltonian circuit does not exist.

Process exited after 10.95 seconds with return value 0
Press any key to continue . . .
```

Resources

Compilation

Compiler paths

C:\Users\kondu\OneDrive\Documents\assignment problem.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

(globals)

Project Classes C:\Users\kondu\OneDrive\Do

```
Enter the number of agents (or tasks): 2
Enter the cost matrix (2 x 2):
2
2
2
2
Minimum cost: 4
Optimal assignment:
Agent 1 -> Task 0
Agent 2 -> Task 0

-----
Process exited after 7.368 seconds with return value 0
Press any key to continue . . . |
```

84 L 85

The screenshot shows a C++ development environment with a code editor and a terminal window.

**Code Editor:**

```
FILE EDIT SEARCH VIEW Project Execute Tools Active Window Help
insert a number.cpp sum of subsets.cpp graph coloring.cpp container loading.cpp Untitled.cpp
Project Classes Debug insert a number.cpp
1 #include <stdio.h>
2
3 void generateFactors(int m) {
4     printf("Factors of %d are: ", m);
5     for (int i = 1; i <= m; ++i) {
6         if (m % i == 0) {
7             printf("%d ", i);
8         }
9     }
10    printf("\n");
11 }
12
13 int main() {
14     int m;
15     printf("Enter a number to find its factors: ");
16     scanf("%d", &m);
17     generateFactors(m);
18     return 0;
19 }
```

**Terminal Output:**

```
C:\Users\kondu\OneDrive\Do ...
Enter a number to find its factors: 2
Factors of 2 are: 1 2
-----
Process exited after 6.89 seconds with return value 0
Press any key to continue . . .
```

```
globals) Debug insert a number.cpp sum of subsets.cpp graph coloring.cpp container loading.cpp Untitled5 n factorial.cpp assignment problem.cpp linear search.cpp [*] Untitled9 hamilton circuit.cpp
53 void hamiltonianCircuit() {
54     path[0] = 0; // Start from vertex 0 as the first vertex in the path
55     for (int i = 1; i < numVertices; i++)
56         path[i] = -1; // Initialize all other vertices as not included in the path
57
58 C:\Users\kondu\OneDrive\Do + -
59
60 Enter number of vertices: 2
61 Enter the adjacency matrix (2 x 2):
62 2
63 2
64 2
65 2
66 Hamiltonian circuit does not exist.
67
68 -----
69 Process exited after 10.95 seconds with return value 0
70 Press any key to continue . . . |
```

Resources

- Warnings: 0

```
Classes Debug insert a number.cpp sum of subsets.cpp graph coloring.cpp container loading.cpp Untitled2 n factorial.cpp assignment problem.cpp binary search.cpp
1 #include <stdio.h>
C:\Users\kondu\OneDrive\Do + v
Element 34 found at index 3.
-----
Process exited after 0.04717 seconds with return value 0
Press any key to continue . . .
```

A screenshot of a C IDE interface. On the left, the code editor shows a file named `copyString.cpp` with the following content:

```
1 // C program to copy the string using
2 // strcpy function
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7
8 // Function to copy the string
9 char* copyString(char s[])
0 {
1     char* s2;
2     s2 = (char*)malloc(20);
3
4     strcpy(s2, s);
5     return (char*)s2;
6 }
7
8 // Driver Code
9 int main()
0 {
1     char s1[20] = "GeeksforGeeks";
2     char* s2;
3
4     // Function Call
5     s2 = copyString(s1);
6     printf("%s", s2);
7     return 0;
8 }
```

The terminal window on the right displays the output of the program:

```
C:\Users\kondu\OneDrive\Do ...
GeeksforGeeks
-----
Process exited after 0.06099 seconds with return value 0
Press any key to continue . . . |
```

```
fibonacci series.cpp | armstrong number.cpp | gcd.cpp | factorial.cpp | prime or not.cpp | sorting order.cpp | bubble sort.cpp | matrix multiplication.cpp | string palindrom or not.cpp
```

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 bool isPalindrome(string str) {
7     int low = 0;
8     int high = str.size() - 1;
9
10    // Keep comparing characters while they are same
11    while (low < high) {
12        if (str[low] != str[high]) {
13            return false; // not a palindrome.
14        }
15        low++; // move the low index forward
16        high--; // move the high index backwards
17    }
18    return true; // is a palindrome
19 }
20
21 int main()
22 {
23     string str= "abbba";
24     string str1 = "abcded";
25
26     cout << str << " is palindrome " << isPalindrome(str) << endl;
27     cout << str1 << " is palindrome " << isPalindrome(str1) << endl;
28     return 0;
29 }
```

```
C:\Users\kondu\OneDrive\Documents\string palindrom or not.exe
abbba is palindrome 1
abcded is palindrome 0
-----
Process exited after 0.04903 seconds with return value 0
Press any key to continue . . .
```

C:\Users\kondu\OneDrive\Documents\matrix multiplication.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

Project Classes Debug fibonacci series.cpp | armstrong number.cpp | gcd.cpp | factorial.cpp | prime or not.cpp

```
18 printf("enter the second matrix element=\n");
19 for(i=0;i<r;i++)
20 {
21     for(j=0;j<c;j++)
22     {
23         scanf("%d",&b[i][j]);
24     }
25 }
26
27 printf("multiply of the matrix=\n");
28 for(i=0;i<r;i++)
29 {
30     for(j=0;j<c;j++)
31     {
32         mul[i][j]=0;
33         for(k=0;k<c;k++)
34         {
35             mul[i][j]+=a[i][k]*b[k][j];
36         }
37     }
38 }
39 //for printing result
40 for(i=0;i<r;i++)
41 {
42     for(j=0;j<c;j++)
43     {
44         printf("%d\t",mul[i][j]);
45     }
46     printf("\n");
47 }
48 return 0;
```

enter the number of row=2  
enter the number of column=2  
enter the first matrix element=1 2  
3 4  
enter the second matrix element=4 5  
6 7  
multiply of the matrix=16 19  
36 43

-----  
Process exited after 11.87 seconds with return value 0  
Press any key to continue . . .

The screenshot shows a C IDE interface with a code editor and a terminal window.

**Code Editor:**

```
1 #include <stdio.h>
2
3 void swap(int* xp, int* yp)
4 {
5     int temp = *xp;
6     *xp = *yp;
7     *yp = temp;
8 }
9
10 void selectionSort(int arr[], int n)
11 {
12     int i, j, min_idx;
13
14     // One by one move boundary of unsorted subarray
15     for (i = 0; i < n - 1; i++) {
16         // Find the minimum element in unsorted array
17         min_idx = i;
18         for (j = i + 1; j < n; j++)
19             if (arr[j] < arr[min_idx])
20                 min_idx = j;
21
22         // Swap the found minimum element with the first
23         // element
24         swap(&arr[min_idx], &arr[i]);
25     }
26 }
27
28 /* Function to print an array */
29 void printArray(int arr[], int size)
30 {
31     int i;
32     for (i = 0; i < size; i++)
```

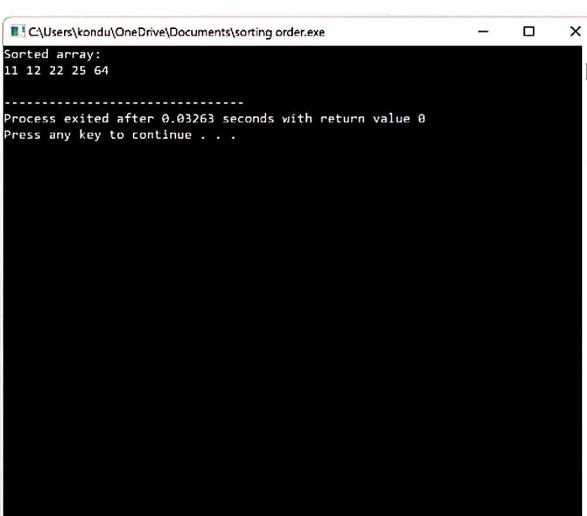
**Terminal Output:**

```
C:\Users\kondu\OneDrive\Documents\bubble sort.exe
Sorted array:
11 12 22 25 64
-----
Process exited after 0.03691 seconds with return value 0
Press any key to continue . . .
```

```
Project Classes Debug fibonacci series.cpp armstrong number.cpp gcd.cpp factorial.cpp prime or not.cpp sorting order.cpp
15 for (i = 0; i < n - 1; i++) {
16     // Find the minimum element in unsorted array
17     min_idx = i;
18     for (j = i + 1; j < n; j++)
19         if (arr[j] < arr[min_idx])
20             min_idx = j;
21
22     // Swap the found minimum element with the first
23     // element
24     swap(&arr[min_idx], &arr[i]);
25 }
26
27
28 /* Function to print an array */
29 void printArray(int arr[], int size)
30 {
31     int i;
32     for (i = 0; i < size; i++)
33         printf("%d ", arr[i]);
34     printf("\n");
35 }
36
37 // Driver program to test above functions
38 int main()
39 {
40     int arr[] = { 64, 25, 12, 22, 11 };
41     int n = sizeof(arr) / sizeof(arr[0]);
42     selectionSort(arr, n);
43     printf("Sorted array: \n");
44     printArray(arr, n);
45     return 0;
46 }
```

C:\Users\kondu\OneDrive\Documents\sorting order.exe  
Sorted array:  
11 12 22 25 64  
-----  
Process exited after 0.03263 seconds with return value 0  
Press any key to continue . . .

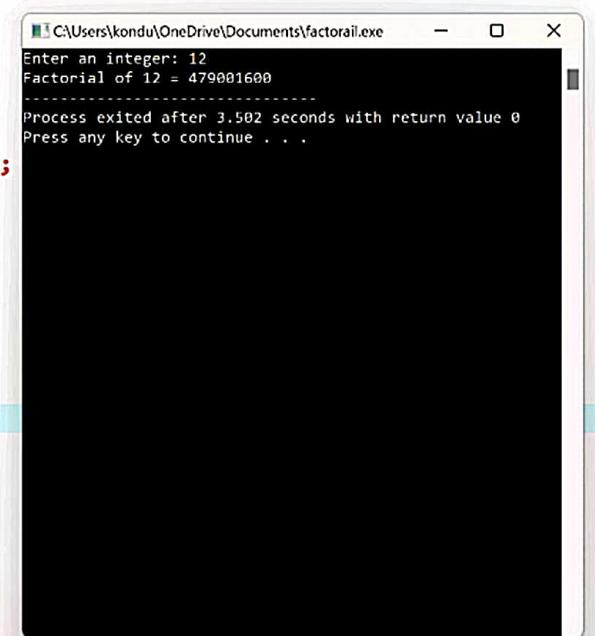
```
Project Classes Debug fibonacci series.cpp armstrong number.cpp gcd.cpp factorial.cpp prime or not.cpp sorting order.cpp
15     for (i = 0; i < n - 1; i++) {
16         // Find the minimum element in unsorted array
17         min_idx = i;
18         for (j = i + 1; j < n; j++)
19             if (arr[j] < arr[min_idx])
20                 min_idx = j;
21
22         // Swap the found minimum element with the first
23         // element
24         swap(&arr[min_idx], &arr[i]);
25     }
26
27
28     /* Function to print an array */
29     void printArray(int arr[], int size)
30 {
31     int i;
32     for (i = 0; i < size; i++)
33         printf("%d ", arr[i]);
34     printf("\n");
35 }
36
37     // Driver program to test above functions
38     int main()
39 {
40     int arr[] = { 64, 25, 12, 22, 11 };
41     int n = sizeof(arr) / sizeof(arr[0]);
42     selectionSort(arr, n);
43     printf("Sorted array: \n");
44     printArray(arr, n);
45     return 0;
46 }
```

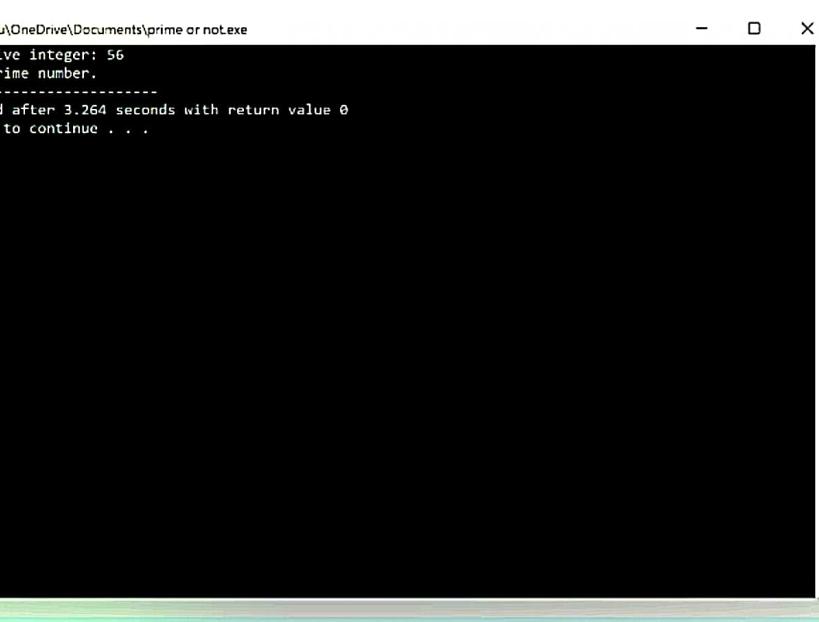


```
#include <stdio.h>
int main() {
    int n, i;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);

    // shows error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
        }
        printf("Factorial of %d = %llu", n, fact);
    }

    return 0;
}
```





```
C:\Users\kondu\OneDrive\Documents> prime or not.exe
Enter a positive integer: 56
56 is not a prime number.

Process exited after 3.264 seconds with return value 0
Press any key to continue . . .
```

The screenshot shows a Windows command prompt window titled "prime or not.exe". The user has entered the command "prime or not.exe" and is prompted to enter a positive integer. They type "56" and the program outputs "56 is not a prime number.". The process exits after 3.264 seconds with a return value of 0, and the user is given the option to press any key to continue.

```
1 #include <stdio.h>
2
3 int main() {
4
5     int n, i, flag = 0;
6     printf("Enter a positive integer: ");
7     scanf("%d", &n);
8
9     // 0 and 1 are not prime numbers
10    // change flag to 1 for non-prime number
11    if (n == 0 || n == 1)
12        flag = 1;
13
14    for (i = 2; i <= n / 2; ++i) {
15
16        // if n is divisible by i, then n is not prime
17        // change flag to 1 for non-prime number
18        if (n % i == 0) {
19            flag = 1;
20            break;
21        }
22    }
23
24    // flag is 0 for prime numbers
25    if (flag == 0)
26        printf("%d is a prime number.", n);
27    else
28        printf("%d is not a prime number.", n);
29
30    return 0;
31}
```

The screenshot shows a C++ development environment with a code editor and a terminal window.

**Code Editor:**

```
max and min.cpp Untitled2.cpp Untitled3 Untitled4.cpp Untitled5.cpp Untitled6.cpp Untitled7.cpp Untitled8.cpp Untitled9.cpp Untitled10.cpp Untitled11.cpp Untitled12.cpp Untitled13.cpp Untitled14.cpp gcd.cpp gcd1.cpp gcd1.cpp
```

```
1 #include <stdio.h>
2 int main()
3 {
4     int n1, n2, i, gcd;
5
6     printf("Enter two integers: ");
7     scanf("%d %d", &n1, &n2);
8
9     for(i=1; i <= n1 && i <= n2; ++i)
10    {
11        // Checks if i is factor of both integers
12        if(n1%i==0 && n2%i==0)
13            gcd = i;
14    }
15
16    printf("G.C.D of %d and %d is %d", n1, n2, gcd);
17
18    return 0;
19 }
```

**Terminal Window:**

```
C:\Users\kondu\OneDrive\Do + X
Enter two integers: 12
16
G.C.D of 12 and 16 is 4
-----
Process exited after 9.343 seconds with return value 0
Press any key to continue . . .
```

\Users\kondu\OneDrive\Documents\gcd.cpp - [Executing] - Dev-C++ 5.11

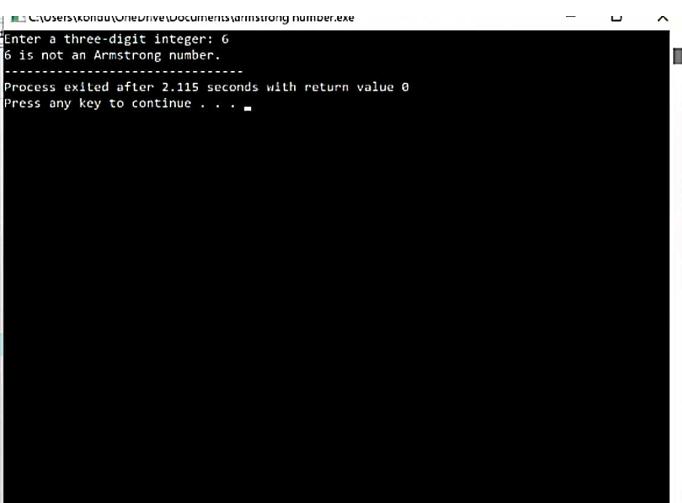
Edit Search View Project Execute Tools AStyle Window Help

File Classes Debug fibonacci series.cpp armstrong number.cpp gcd.cpp

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     double arr[100];
5     printf("Enter the number of elements (1 to 100): ");
6     scanf("%d", &n);
7
8     for (int i = 0; i < n; ++i) {
9         printf("Enter number%d: ", i + 1);
10        scanf("%lf", &arr[i]);
11    }
12
13     // storing the Largest number to arr[0]
14    for (int i = 1; i < n; ++i) {
15        if (arr[0] < arr[i]) {
16            arr[0] = arr[i];
17        }
18    }
19
20    printf("Largest element = %.2lf", arr[0]);
21
22    return 0;
23 }
```

C:\Users\kondu\OneDrive\Documents\gcd.exe

```
Enter the number of elements (1 to 100): 3
Enter number1: 24
Enter number2: 29
Enter number3: 23
Largest element = 29.00
-----
Process exited after 8.884 seconds with return value 0
Press any key to continue . . .
```



```
Project Classes Debug fibonacci series.cpp armstrong number.cpp TIEM-GCC 4.9.2 E4-Build (qicba1s) C:\Users\xoraut\OneDrive\Documents\armstrong number.exe Enter a three-digit integer: 6 6 is not an Armstrong number. ----- Process exited after 2.115 seconds with return value 0 Press any key to continue . . .
```

The screenshot shows a code editor and a terminal window. The code editor has tabs for 'fibonacci series.cpp' and 'armstrong number.cpp'. The 'armstrong number.cpp' tab is active, displaying the following C++ code:

```
1 #include <stdio.h>
2 int main() {
3     int num, originalNum, remainder, result = 0;
4     printf("Enter a three-digit integer: ");
5     scanf("%d", &num);
6     originalNum = num;
7
8     while (originalNum != 0) {
9         // remainder contains the last digit
10        remainder = originalNum % 10;
11
12        result += remainder * remainder * remainder;
13
14        // removing last digit from the original number
15        originalNum /= 10;
16    }
17
18    if (result == num)
19        printf("%d is an Armstrong number.", num);
20    else
21        printf("%d is not an Armstrong number.", num);
22
23    return 0;
24 }
```

The terminal window to the right shows the execution of the program. It prompts the user to enter a three-digit integer (6), displays the result '6 is not an Armstrong number.', and then exits.

```
fibonacci series.cpp
1 #include <stdio.h>
2
3 // Function to calculate the nth Fibonacci number
4 int fibonacci(int n) {
5     if (n <= 1) {
6         return n;
7     }
8     return fibonacci(n - 1) + fibonacci(n - 2);
9 }
10
11 // Function to print the Fibonacci series up to nth term
12 void printFibonacci(int n) {
13     printf("Fibonacci series up to %d terms: ", n);
14     for (int i = 0; i < n; i++) {
15         printf("%d ", fibonacci(i));
16     }
17     printf("\n");
18 }
19
20 int main() {
21     int n;
22     printf("Enter the number of terms: ");
23     scanf("%d", &n);
24     printFibonacci(n);
25     return 0;
26 }
```

```
C:\Users\kondu\OneDrive\Documents\fibonacci series.exe
Enter the number of terms: 5
Fibonacci series up to 5 terms: 0 1 1 2 3
-----
Process exited after 2.13 seconds with return value 0
Press any key to continue . . .
```