

# Day -1

## Assignment

Menda Mani Sai

192111399

### 1. Write a program to Print Fibonacci Series using recursion.

```
#include<stdio.h>

int fibbno(int n)

{
    if(n==0)

        return 0;

    else if(n==1)

        return 1;

    else

        return fibbno(n - 1) + fibbno(n- 2);
}

int main()

{

    int n,i;

    printf("Enter the range =");

    scanf("%d", &n);

    for(i=0;i<n;i++)
```

```

{
printf("%d",fibbno(i));

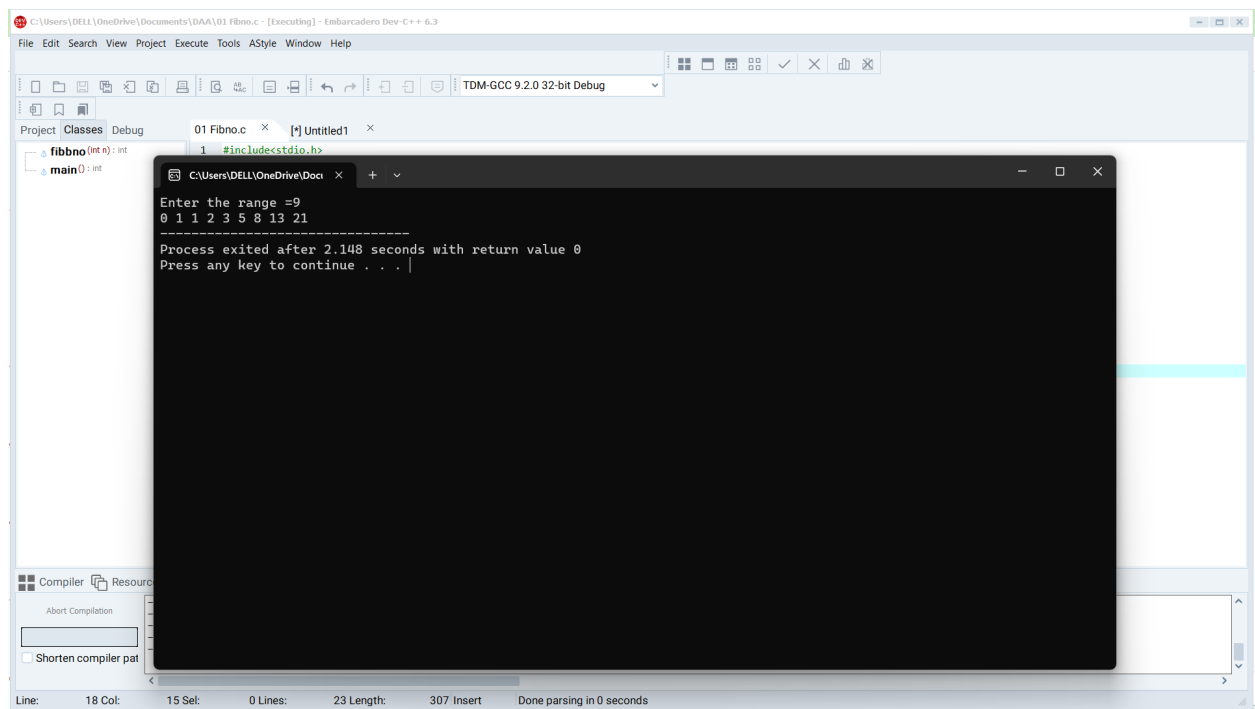
}

return 0;

}

```

## OUTPUT



2. Write a program to check the given no is Armstrong or not. .

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

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

```
int main()
```

```
{
```

```
int n,r,sum=0,temp;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{

r=n%10;

sum=sum+(r*r*r);

n=n/10;

}

if(temp==sum)

printf("armstrong number ");

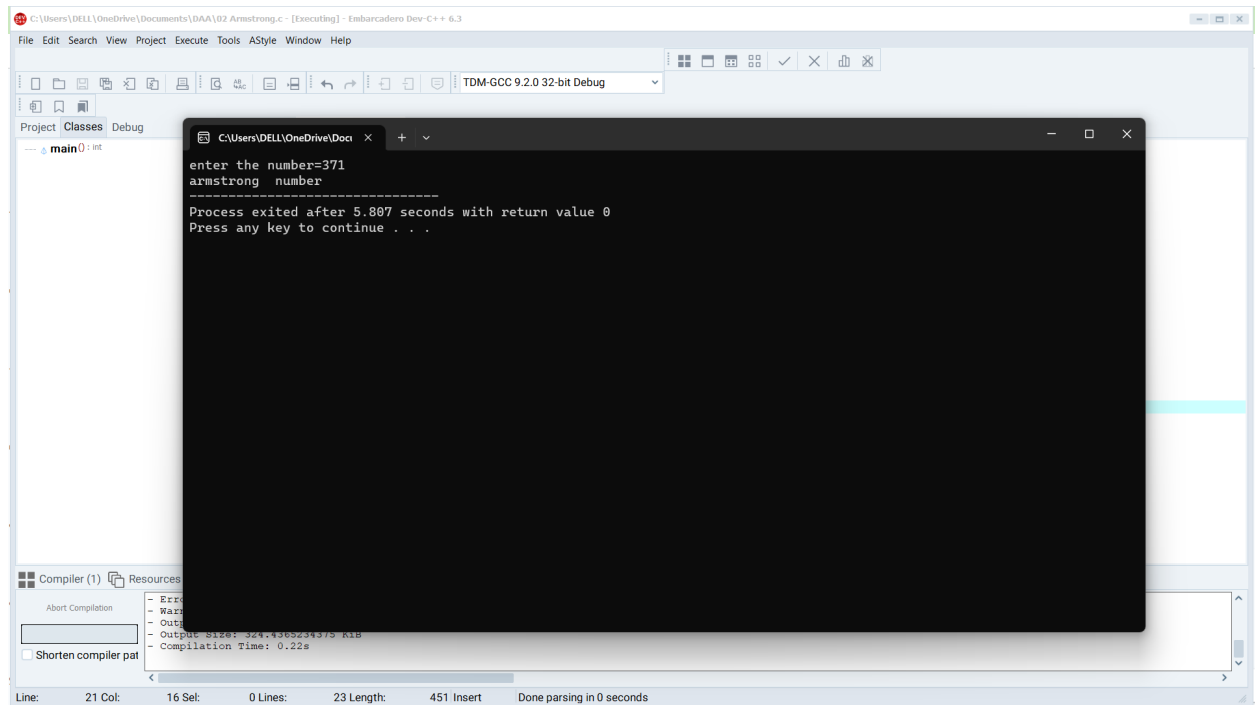
else

printf("not armstrong number");

return 0;

}
```

# OUTPUT



## 3. Write a program to find the GCD of two numbers.

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

```
int main()
```

```
{
```

```
    int n1, n2, i, gcd;
```

```
    printf("Enter two integers: ");
```

```
    scanf("%d %d", &n1, &n2);
```

```
    for(i=1; i <= n1 && i <= n2; ++i)
```

```
    {
```

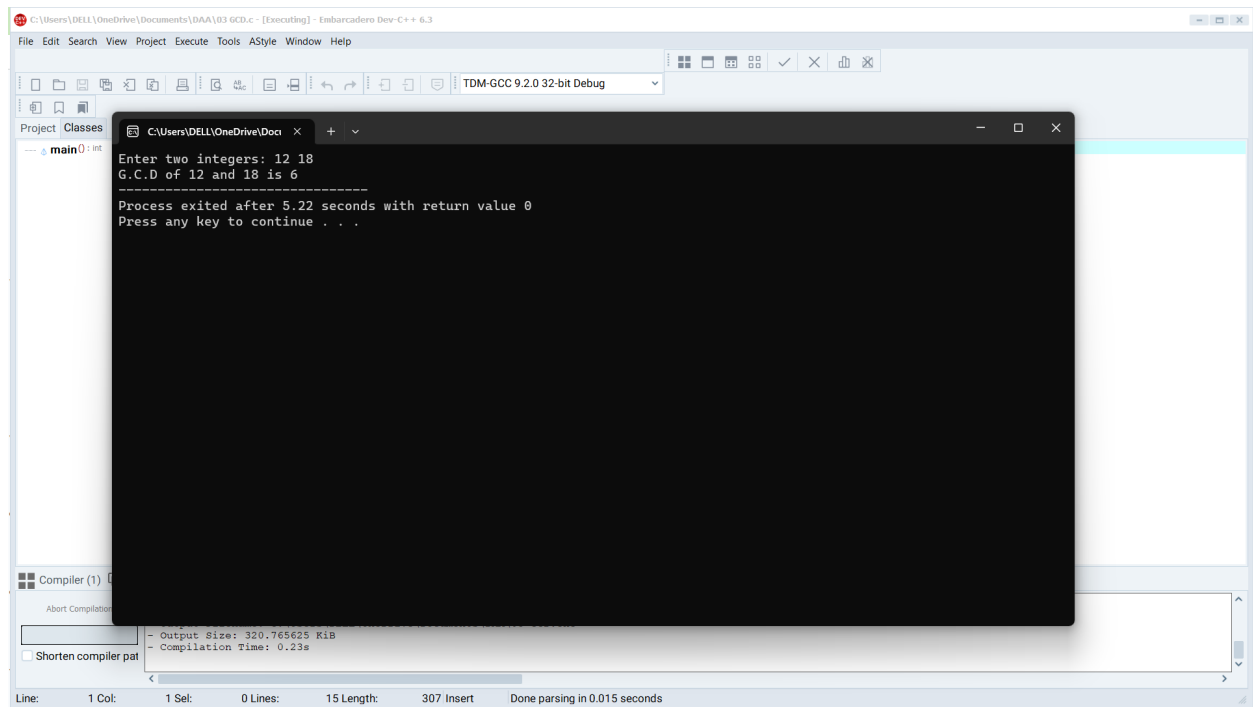
```
        if(n1%i==0 && n2%i==0)
```

```
        gcd = i;
```

```
    }
```

```
printf("G.C.D of %d and %d is %d", n1, n2, gcd);  
  
return 0;  
  
}
```

## OUTPUT

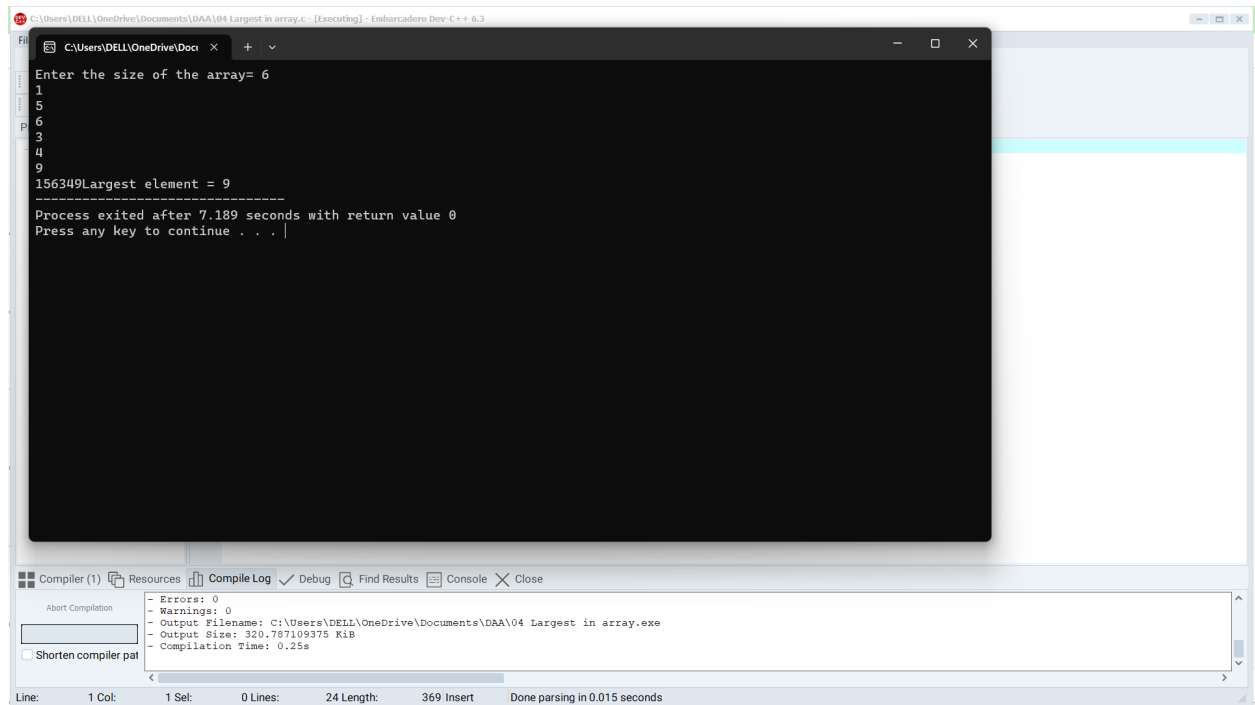


## 4. Write a program to get the largest element of an array.

```
#include<stdio.h>  
  
int main()  
{  
    int a[1000];  
  
    int i,n;  
  
    printf("Enter the size of the array= ");  
  
    scanf("%d",&n);
```

```
    for (i=0;i<n;i++)  
{  
    scanf("%d", &a[i]);  
}  
  
    for (i=0;i<n;i++){  
    printf("%d", a[i]);  
}  
  
    for (int i = 1; i < n; ++i) {  
        if (a[0] < a[i]) {  
            a[0] = a[i];  
        }  
    }  
  
    printf("Largest element = %d", a[0]);  
return 0;  
}
```

# OUTPUT



The screenshot shows a C++ IDE with a console window displaying the output of a program. The program prompts the user to enter the size of the array, which is 6. It then displays a list of numbers: 1, 5, 6, 3, 4, 9. The output shows the largest element is 9. The console also displays the process exit message: "Process exited after 7.189 seconds with return value 0" and "Press any key to continue . . . |". The IDE interface includes a compiler window showing 0 errors and 0 warnings, and a status bar at the bottom indicating the current line and column.

```
Enter the size of the array= 6
1
5
6
3
4
9
156349Largest element = 9
-----
Process exited after 7.189 seconds with return value 0
Press any key to continue . . . |
```

## 5. Write a program to find the Factorial of a number.

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

```
int fact(int a)
```

```
{
```

```
    if(a==0)
```

```
        return 1;
```

```
    else if(a==1)
```

```
        return 1;
```

```
    else
```

```
        return (a*fact(a-1));
```

```
}
```

```
int main()
```

```

{
    int a,i;

    printf("Enter the number to know the factorial= ");

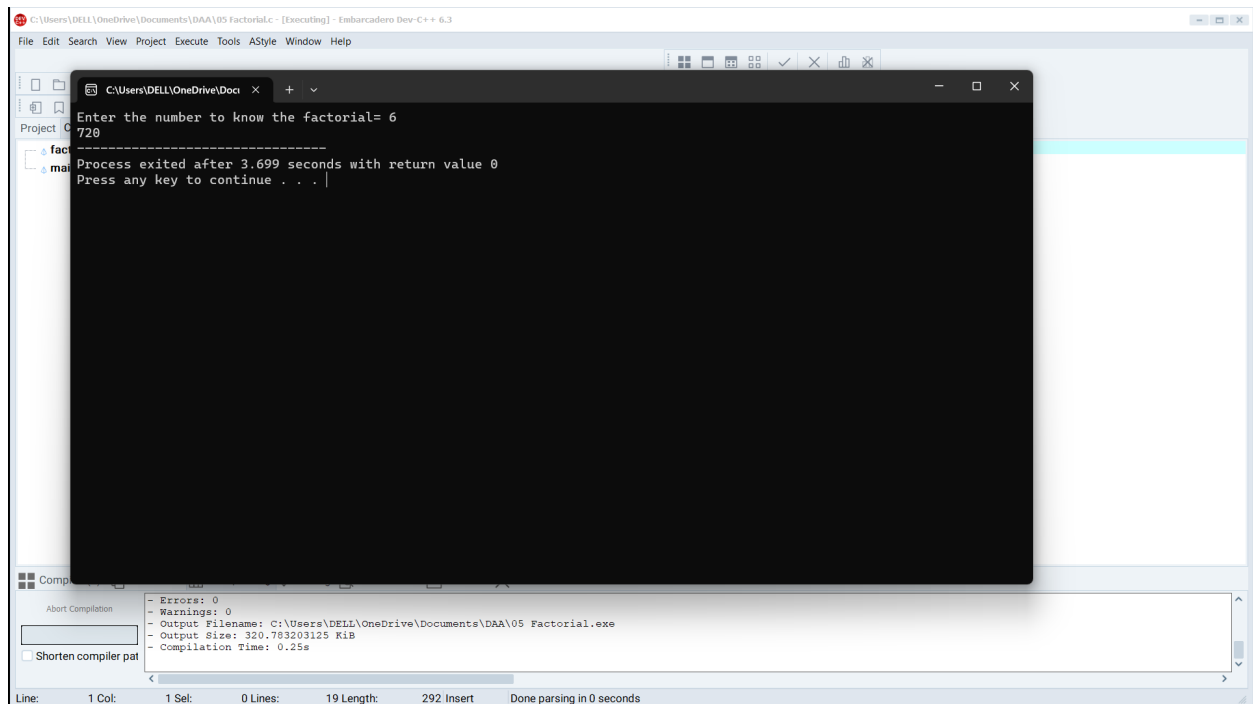
    scanf("%d", &a);

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

    return 0;
}

```

## OUTPUT



```

C:\Users\DELL\OneDrive\Documents\DAA\05 Factorial.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help

C:\Users\DELL\OneDrive\Doc...
+ -
- - -
Enter the number to know the factorial= 6
720
-----
Process exited after 3.699 seconds with return value 0
Press any key to continue . . . |

Comp...
Abort Compilation
Shorten compiler pat...
Errors: 0
Warnings: 0
Output Filename: C:\Users\DELL\OneDrive\Documents\DAA\05 Factorial.exe
Output Size: 320,783,031,25 KiB
Compilation Time: 0.25s
Line: 1 Col: 1 Sel: 0 Lines: 19 Length: 292 Insert Done parsing in 0 seconds

```

6. Write a program to check a number is a prime number or not.

```

#include<stdio.h>

int main()

{

```



```
int a,c=0;

printf("Enter a number= ");

scanf("%d", &a);

for(int i=1;i<=a;i++)

{

    if(a%i==0){

        c++;

    }

}

if(c==2)

    printf("The number %d is prime",a);

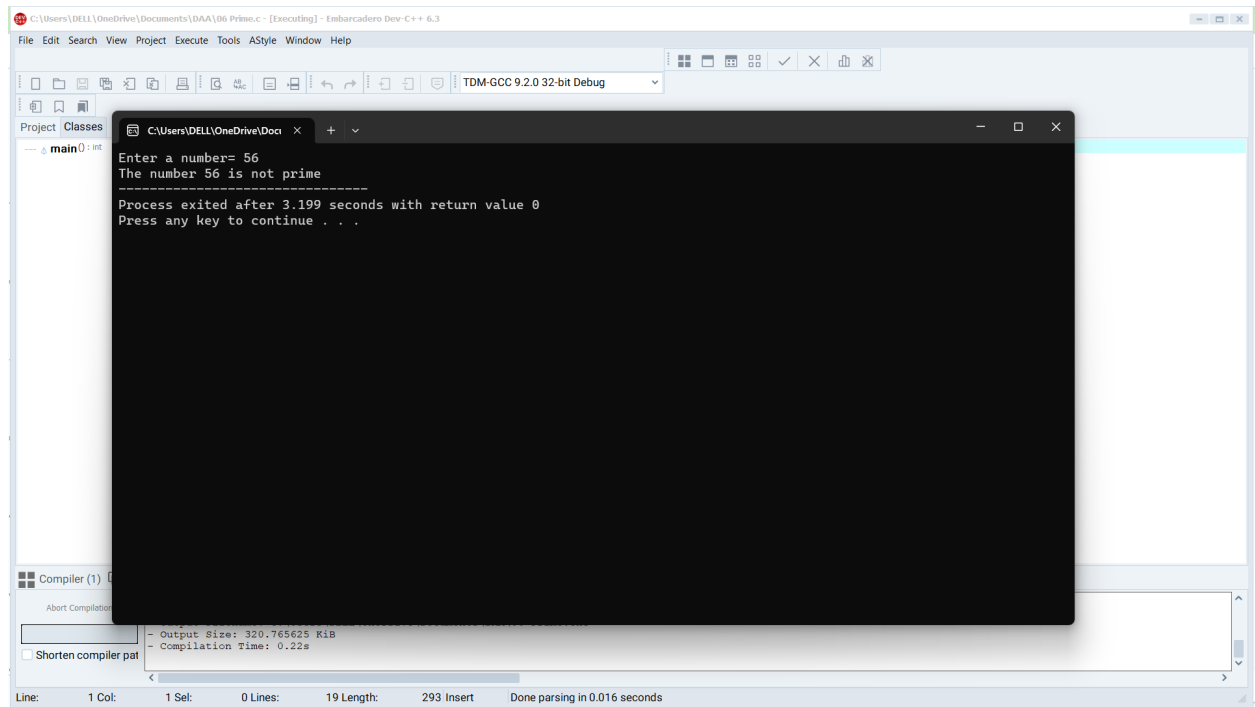
else

    printf("The number %d is not prime",a);

return 0;

}
```

# OUTPUT



## 7. Write a program to perform Selection sort.

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

```
void selectionSort(int arr[], int n) {  
    for (int i = 0; i < n - 1; i++) {  
        int minIndex = i;  
        for (int j = i + 1; j < n; j++) {  
            if (arr[j] < arr[minIndex]) {  
                minIndex = j;  
            }  
        }  
        int temp = arr[minIndex];  
        arr[minIndex] = arr[i];  
    }  
}
```

```
        arr[i] = temp;
    }
}

int main() {
    int n;

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

    scanf("%d", &n);

    int arr[n];

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

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

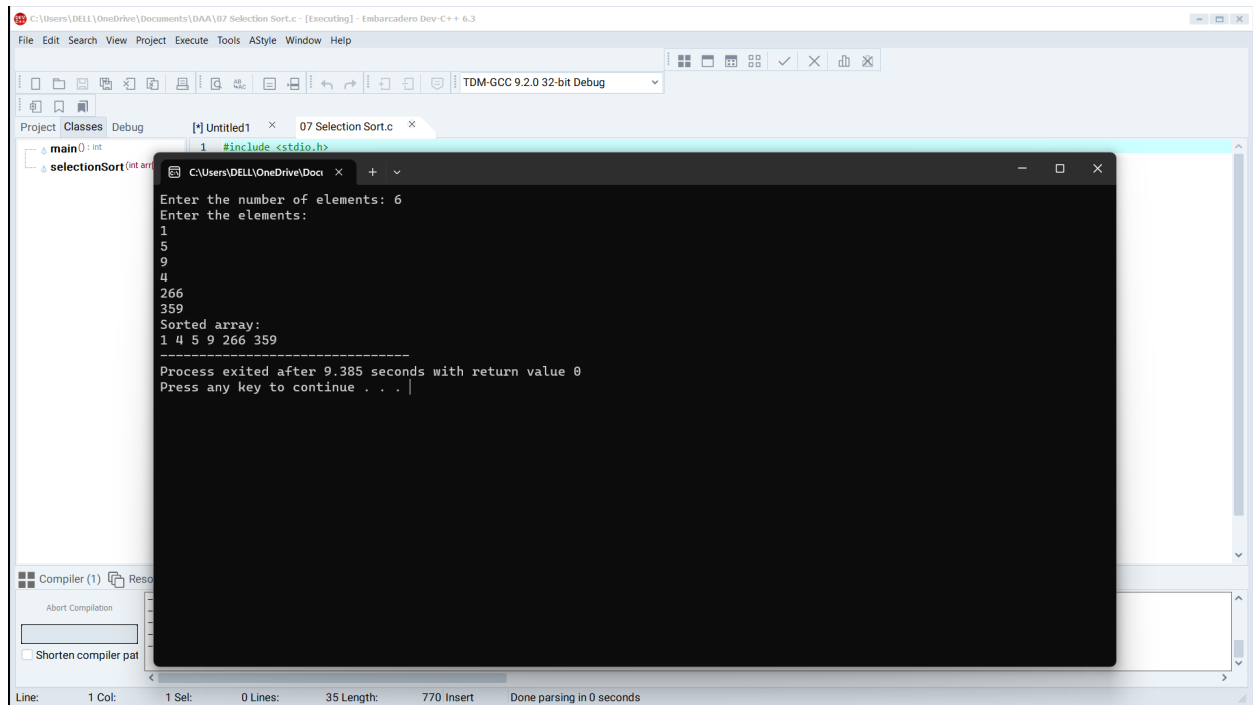
    selectionSort(arr, n);

    printf("Sorted array:\n");

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

    return 0;
}
```

# OUTPUT



```
Enter the number of elements: 6
Enter the elements:
1
5
9
4
266
359
Sorted array:
1 4 5 9 266 359
-----
Process exited after 9.385 seconds with return value 0
Press any key to continue . . .
```

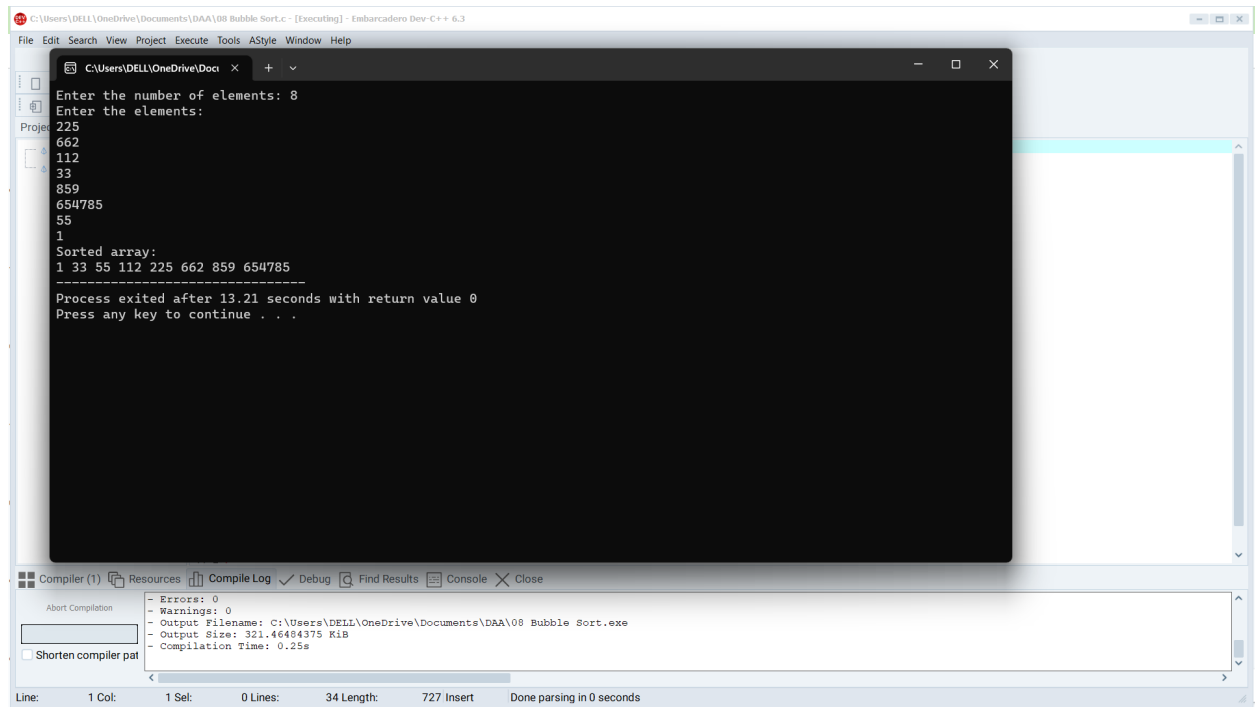
## 8. Write a program to perform Bubble sort

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

```
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
```

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

# OUTPUT



```
Enter the number of elements: 8
Enter the elements:
225
662
112
32
859
654785
55
1
Sorted array:
1 32 55 112 225 662 859 654785
-----
Process exited after 13.21 seconds with return value 0
Press any key to continue . . .
```

## 9. Write a program for to multiply two Matrixes

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

```
void matrix_multiply(int a[][100], int b[][100], int result[][100], int rows_a, int cols_a, int cols_b) {
    for (int i = 0; i < rows_a; i++) {
        for (int j = 0; j < cols_b; j++) {
            result[i][j] = 0;
            for (int k = 0; k < cols_a; k++) {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```

```
}
```

```
void print_matrix(int matrix[][100], int rows, int cols) {
```

```
    for (int i = 0; i < rows; i++) {
```

```
        for (int j = 0; j < cols; j++) {
```

```
            printf("%d ", matrix[i][j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
int main() {
```

```
    int rows_a, cols_a, rows_b, cols_b;
```

```
    printf("Enter the dimensions of matrix A (rows columns): ");
```

```
    scanf("%d %d", &rows_a, &cols_a);
```

```
    printf("Enter the dimensions of matrix B (rows columns): ");
```

```
    scanf("%d %d", &rows_b, &cols_b);
```

```
    if (cols_a != rows_b) {
```

```
        printf("Matrix multiplication not possible due to incompatible dimensions.\n");
```

```
        return 1;
```

```
    }
```

```
int matrixA[100][100];
```

```
int matrixB[100][100];
```

```
printf("Enter the elements of matrix A:\n");
```

```
for (int i = 0; i < rows_a; i++) {
```

```
    for (int j = 0; j < cols_a; j++) {
```

```
        scanf("%d", &matrixA[i][j]);
```

```
    }
```

```
}
```

```
printf("Enter the elements of matrix B:\n");
```

```
for (int i = 0; i < rows_b; i++) {
```

```
    for (int j = 0; j < cols_b; j++) {
```

```
        scanf("%d", &matrixB[i][j]);
```

```
    }
```

```
}
```

```
int resultMatrix[100][100];
```

```
matrix_multiply(matrixA, matrixB, resultMatrix, rows_a, cols_a, cols_b);
```

```
printf("\nMatrix A:\n");
```

```
print_matrix(matrixA, rows_a, cols_a);
```



```

printf("\nMatrix B:\n");

print_matrix(matrixB, rows_b, cols_b);


printf("\nResult Matrix:\n");

print_matrix(resultMatrix, rows_a, cols_b);


return 0;

}

```

## OUTPUT

```

C:\Users\DELL\OneDrive\Documents\09 Matrix Mult.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 32-bit Debug
Project Classes Debug
main0: int
matrix_multiply(int a[100][100], int b[100][100])
print_matrix(int matrix[100][100], int rows, int cols)
37 int matrixA[100][100];
38 int matrixB[100][100];
39
C:\Users\DELL\OneDrive\Documents\09 Matrix Mult.c
Enter the dimensions of matrix B (rows columns): 3
3
Enter the elements of matrix A:
1 2 3
1 2 3
1 2 3
Enter the elements of matrix B:
1 1 1
1 1 1
1 1 1
Matrix A:
1 2 3
1 2 3
1 2 3
Matrix B:
1 1 1
1 1 1
1 1 1
Result Matrix:
6 6 6
6 6 6
6 6 6
-----
Process exited after 24.76 seconds with return value 0
Press any key to continue . . .
Line: 68 Col: 1 Sel: 0 Lines: 68 Length: 1846 Insert Done parsing in 0.016 seconds

```

## 10. Write a program for to check whether a given String is Palindrome or not

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

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

```
#include <string.h>

bool isPalindrome(const char str[]) {
    int length = strlen(str);

    for (int i = 0; i < length / 2; i++) {
        if (str[i] != str[length - 1 - i]) {
            return false;
        }
    }

    return true;
}

int main() {
    char str[100];

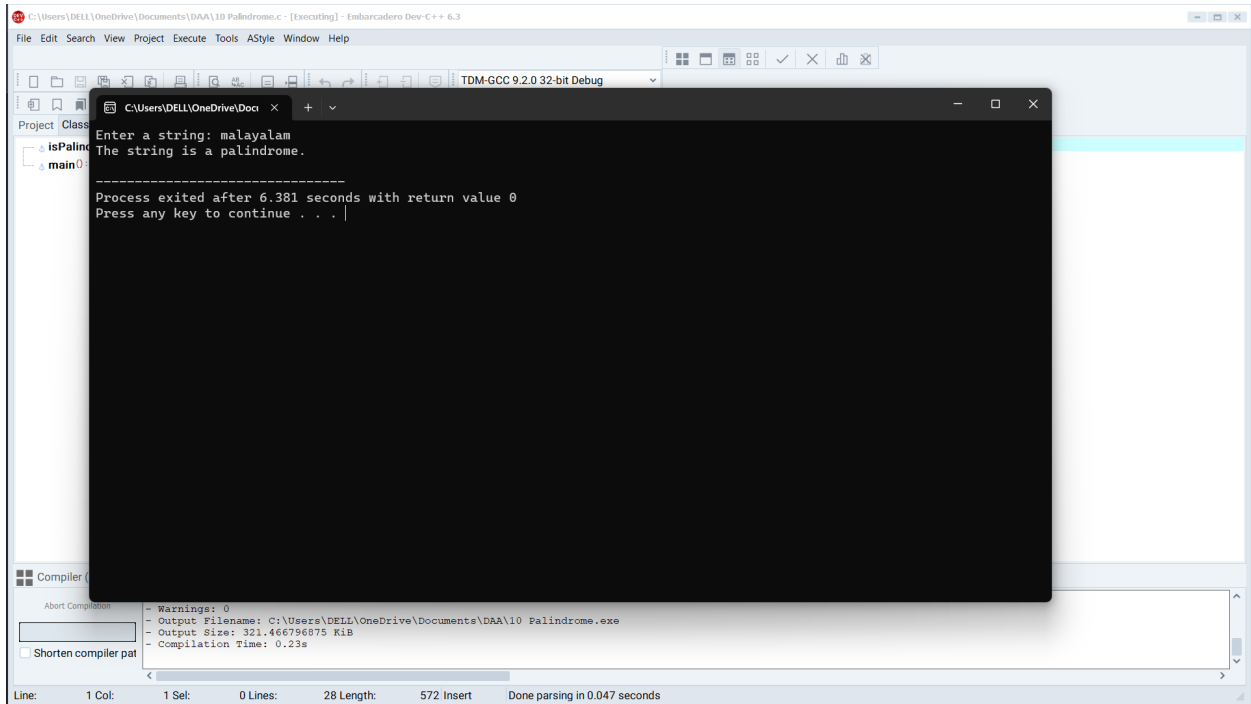
    printf("Enter a string: ");
    scanf("%s", str);

    if (isPalindrome(str)) {
        printf("The string is a palindrome.\n");
    } else {
        printf("The string is not a palindrome.\n");
    }

    return 0;
}
```

```
}
```

## OUTPUT



## 11. Write a program for to copy one string to another

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

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

```
int main()
```

```
{
```

```
    char str1[20], str2[20];
```

```
    printf("Enter the string: ");
```

```
    scanf("%s", str1);
```

```
    printf("\nString 1 = %s", str1);
```

```
    strcpy(str2, str1);
```

```
    printf("\nString 2 = %s", str2);
```

```

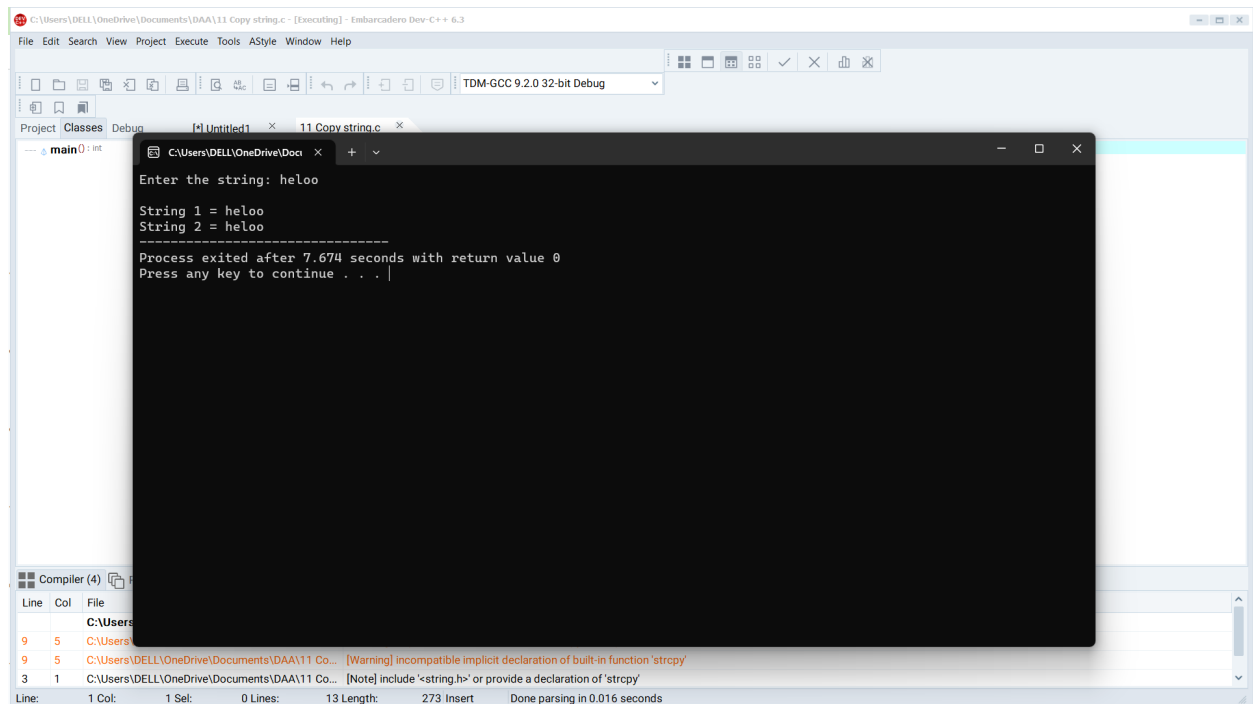
    getch();

    return 0;

}

```

## OUTPUT



## 12. Write a Program to perform binary search.

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

```
int binarySearch(int arr[], int n, int target) {
```

```
    int left = 0;
```

```
    int right = n - 1;
```

```
    while (left <= right) {
```

```
        int mid = left + (right - left) / 2;
```

```
    if (arr[mid] == target) {  
        return mid; // Element found at index 'mid'  
    } else if (arr[mid] < target) {  
        left = mid + 1; // Search the right half  
    } else {  
        right = mid - 1; // Search the left half  
    }  
}  
  
return -1; // Element not found  
}
```

```
int main() {  
    int n;  
    printf("Enter the number of elements: ");  
    scanf("%d", &n);  
  
    int arr[n];  
    printf("Enter the elements in sorted order:\n");  
    for (int i = 0; i < n; i++) {  
        scanf("%d", &arr[i]);  
    }
```

```
    int target;

    printf("Enter the element to search for: ");

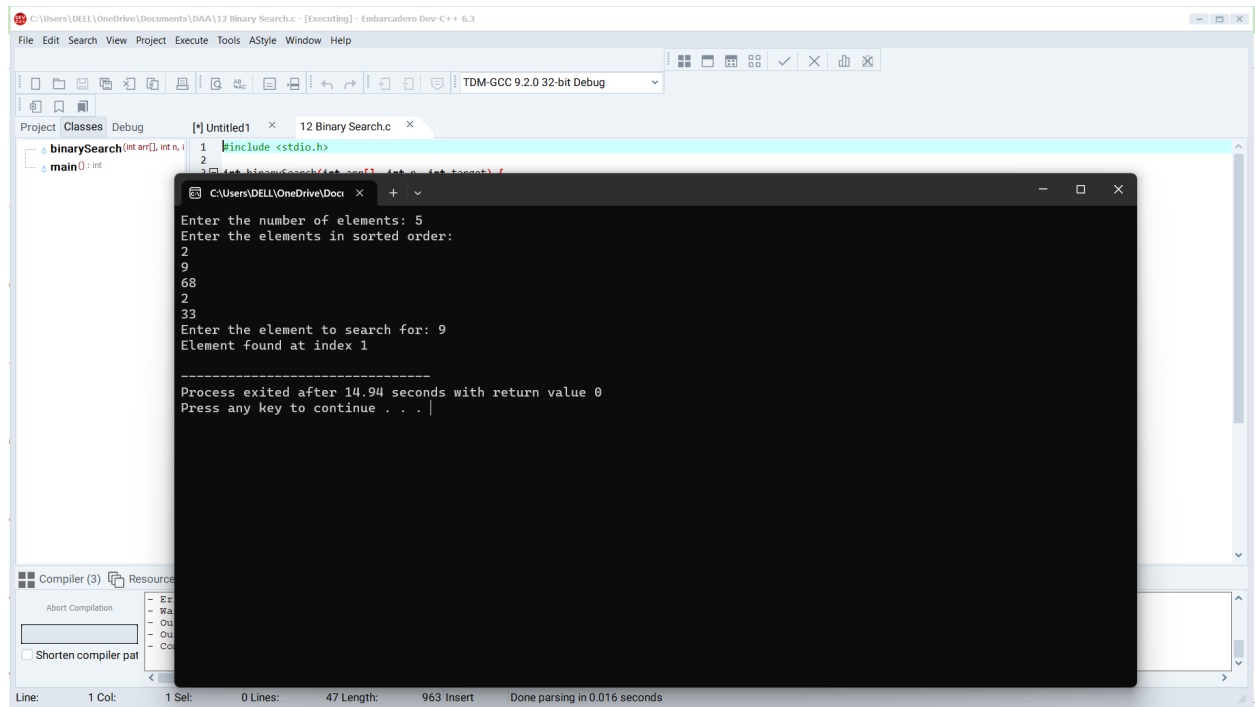
    scanf("%d", &target);


    int index = binarySearch(arr, n, target);


    if (index != -1) {
        printf("Element found at index %d\n", index);
    } else {
        printf("Element not found\n");
    }


    return 0;
}
```

# OUTPUT



```
Enter the number of elements: 5
Enter the elements in sorted order:
2
9
68
2
33
Enter the element to search for: 9
Element found at index 1

-----
Process exited after 14.94 seconds with return value 0
Press any key to continue . . .
```

## 13. Write a program to print the reverse of a string

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

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

```
int main()
```

```
{
```

```
    char Str[1000];
```

```
    int i;
```

```
    printf("Enter the String: ");
```

```
    scanf("%s", Str);
```

```

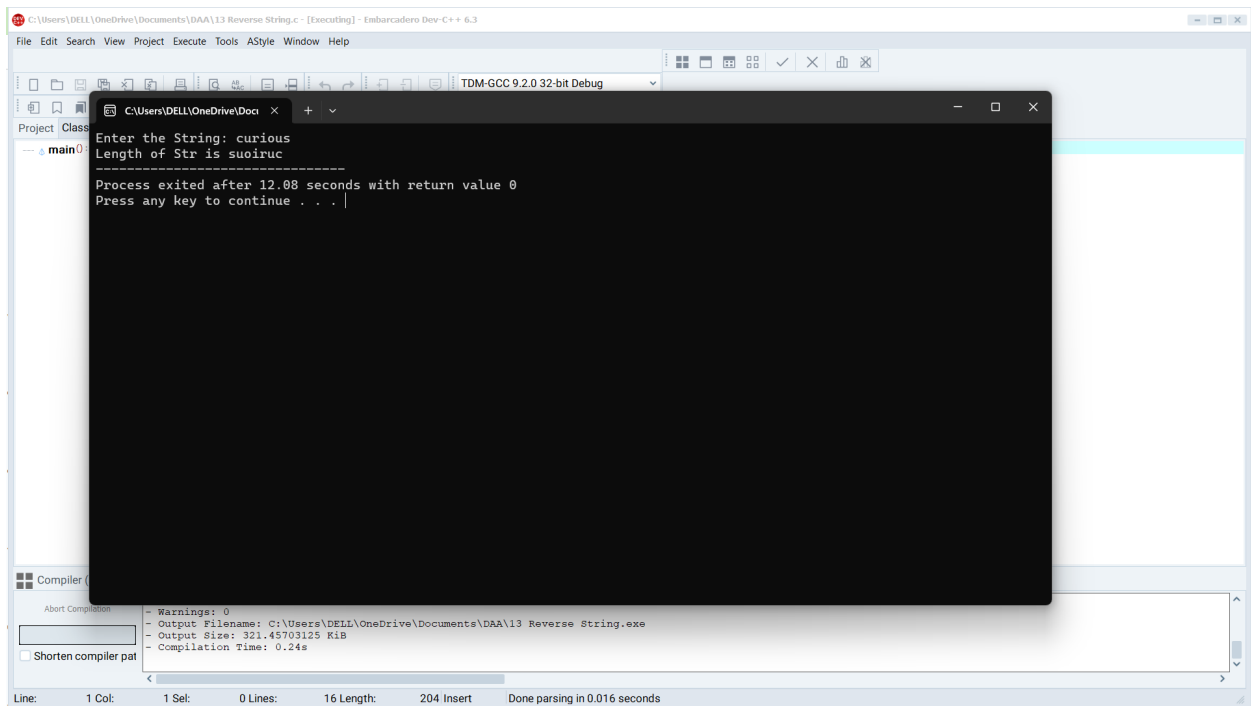
printf("Length of Str is %s", strrev(Str));

return 0;

}

```

## OUTPUT



## 14. Write a program to find the length of a string.

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

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

```
int main()
```

```
{
```

```
    char Str[1000];
```

```
    int i;
```



```

printf("Enter the String: ");

scanf("%s", Str);

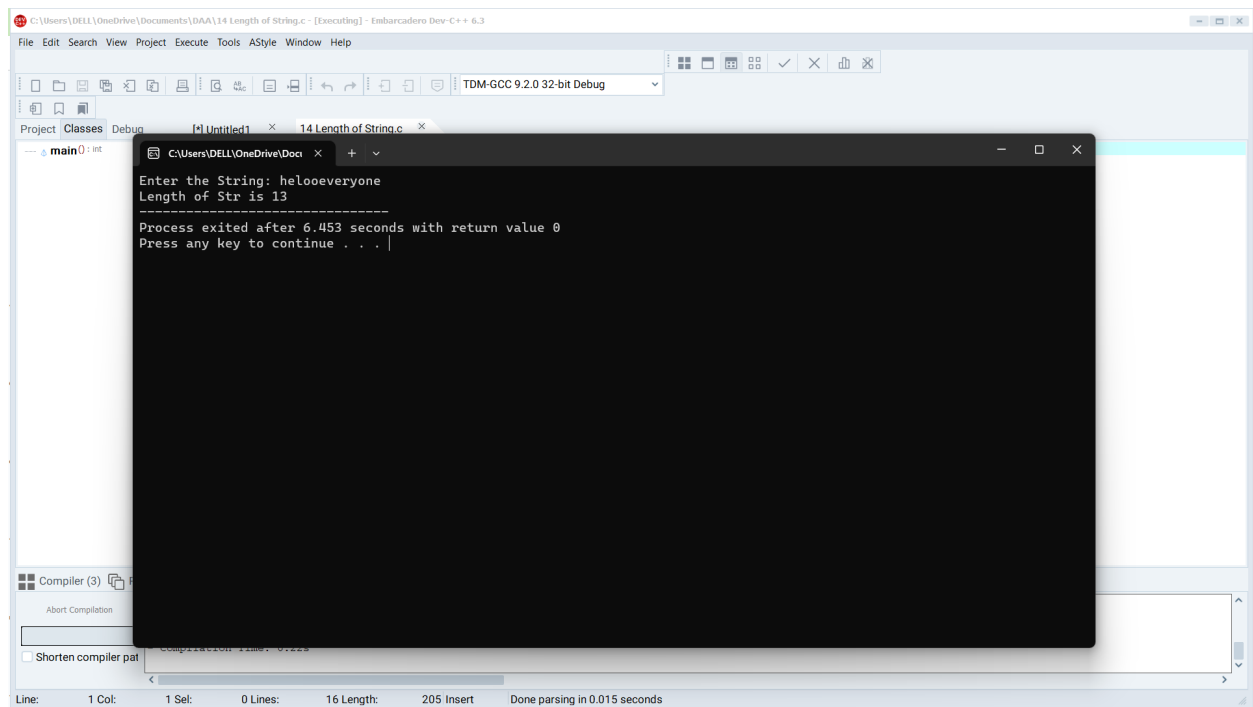

printf("Length of Str is %ld", strlen(Str));


return 0;

}

```

## OUTPUT



The screenshot shows a C++ IDE with a project named "14 Length of String.c". The IDE is running on "Embarcadero Dev-C++ 6.3" using "TDM-GCC 9.2.0 32-bit Debug". The output window displays the following text:

```

Enter the String: helooeveryone
Length of Str is 13
-----
Process exited after 6.453 seconds with return value 0
Press any key to continue . . .

```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing "Line: 1 Col: 1 Sel: 0 Lines: 16 Length: 205 Insert Done parsing in 0.015 seconds".

## 15. Write a program to perform Strassen's Matrix Multiplication.

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

```
int main(){
```

```
int a[2][2], b[2][2], c[2][2], i, j;
```

```
int m1, m2, m3, m4 , m5, m6, m7;
```

```
printf("Enter the 4 elements of first matrix: ");
```

```
for(i = 0; i < 2; i++)
```

```
    for(j = 0; j < 2; j++)
```

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

```
printf("Enter the 4 elements of second matrix: ");
```

```
for(i = 0; i < 2; i++)
```

```
    for(j = 0; j < 2; j++)
```

```
        scanf("%d", &b[i][j]);
```

```
printf("\nThe first matrix is\n");
```

```
for(i = 0; i < 2; i++){
```

```
    printf("\n");
```

```
        for(j = 0; j < 2; j++)
```

```
            printf("%d\t", a[i][j]);
```

```
}
```

```
printf("\nThe second matrix is\n");
```

```
for(i = 0; i < 2; i++){
```

```
    printf("\n");
```

```
        for(j = 0; j < 2; j++)
```

```

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

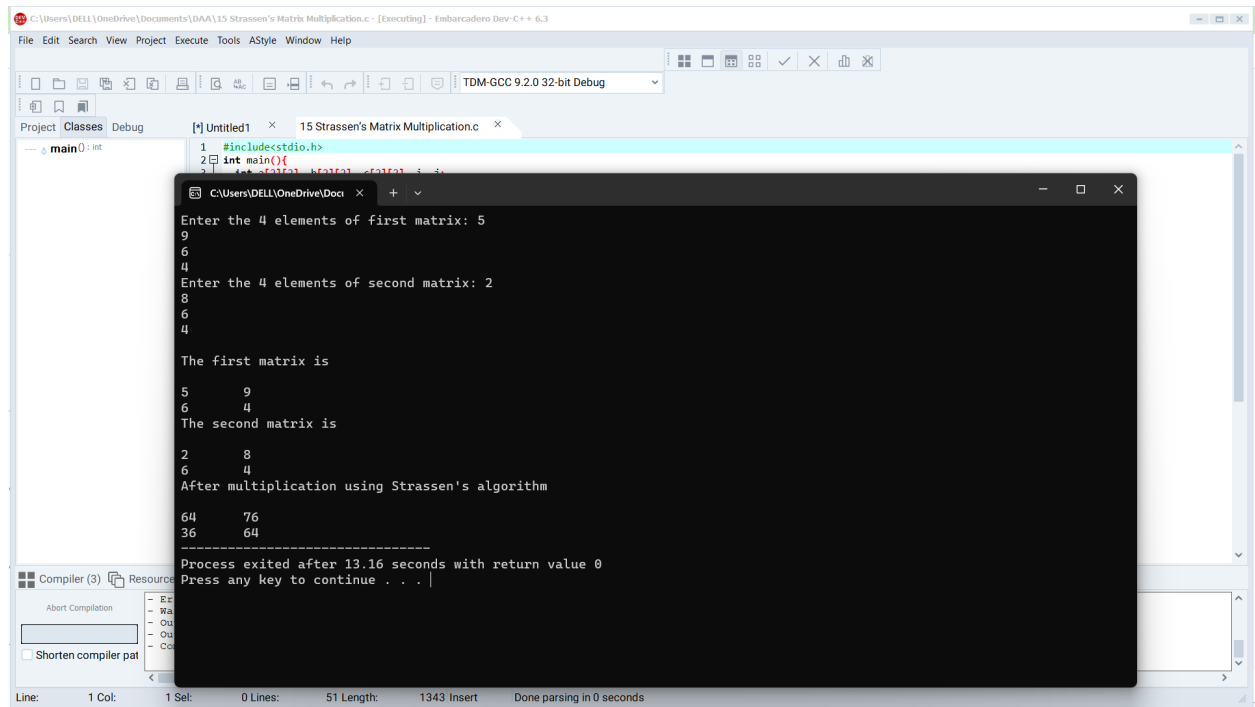
    m1= (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
    m2= (a[1][0] + a[1][1]) * b[0][0];
    m3= a[0][0] * (b[0][1] - b[1][1]);
    m4= a[1][1] * (b[1][0] - b[0][0]);
    m5= (a[0][0] + a[0][1]) * b[1][1];
    m6= (a[1][0] - a[0][0]) * (b[0][0]+b[0][1]);
    m7= (a[0][1] - a[1][1]) * (b[1][0]+b[1][1]);

    c[0][0] = m1 + m4- m5 + m7;
    c[0][1] = m3 + m5;
    c[1][0] = m2 + m4;
    c[1][1] = m1 - m2 + m3 + m6;

    printf("\nAfter multiplication using Strassen's algorithm \n");
    for(i = 0; i < 2 ; i++){
        printf("\n");
        for(j = 0;j < 2; j++)
            printf("%d\t", c[i][j]);
    }
    return 0;
}

```

# OUTPUT



```
1 #include<stdio.h>
2 int main()
3 {
4     int a[2][2], b[2][2], c[2][2];
5     int i, j, k;
6     int n1, n2;
7     int m1, m2;
8     int m3, m4;
9     int m5, m6;
10    int m7, m8;
11    int m9, m10;
12    int m11, m12;
13    int m13, m14;
14    int m15, m16;
15    int m17, m18;
16    int m19, m20;
17    int m21, m22;
18    int m23, m24;
19    int m25, m26;
20    int m27, m28;
21    int m29, m30;
22    int m31, m32;
23    int m33, m34;
24    int m35, m36;
25    int m37, m38;
26    int m39, m40;
27    int m41, m42;
28    int m43, m44;
29    int m45, m46;
30    int m47, m48;
31    int m49, m50;
32    int m51, m52;
33    int m53, m54;
34    int m55, m56;
35    int m57, m58;
36    int m59, m60;
37    int m61, m62;
38    int m63, m64;
39    int m65, m66;
40    int m67, m68;
41    int m69, m70;
42    int m71, m72;
43    int m73, m74;
44    int m75, m76;
45    int m77, m78;
46    int m79, m80;
47    int m81, m82;
48    int m83, m84;
49    int m85, m86;
50    int m87, m88;
51    int m89, m90;
52    int m91, m92;
53    int m93, m94;
54    int m95, m96;
55    int m97, m98;
56    int m99, m100;
57    int m101, m102;
58    int m103, m104;
59    int m105, m106;
60    int m107, m108;
61    int m109, m110;
62    int m111, m112;
63    int m113, m114;
64    int m115, m116;
65    int m117, m118;
66    int m119, m120;
67    int m121, m122;
68    int m123, m124;
69    int m125, m126;
70    int m127, m128;
71    int m129, m130;
72    int m131, m132;
73    int m133, m134;
74    int m135, m136;
75    int m137, m138;
76    int m139, m140;
77    int m141, m142;
78    int m143, m144;
79    int m145, m146;
80    int m147, m148;
81    int m149, m150;
82    int m151, m152;
83    int m153, m154;
84    int m155, m156;
85    int m157, m158;
86    int m159, m160;
87    int m161, m162;
88    int m163, m164;
89    int m165, m166;
90    int m167, m168;
91    int m169, m170;
92    int m171, m172;
93    int m173, m174;
94    int m175, m176;
95    int m177, m178;
96    int m179, m180;
97    int m181, m182;
98    int m183, m184;
99    int m185, m186;
100   int m187, m188;
101   int m189, m190;
102   int m191, m192;
103   int m193, m194;
104   int m195, m196;
105   int m197, m198;
106   int m199, m200;
107   int m201, m202;
108   int m203, m204;
109   int m205, m206;
110   int m207, m208;
111   int m209, m210;
112   int m211, m212;
113   int m213, m214;
114   int m215, m216;
115   int m217, m218;
116   int m219, m220;
117   int m221, m222;
118   int m223, m224;
119   int m225, m226;
120   int m227, m228;
121   int m229, m230;
122   int m231, m232;
123   int m233, m234;
124   int m235, m236;
125   int m237, m238;
126   int m239, m240;
127   int m241, m242;
128   int m243, m244;
129   int m245, m246;
130   int m247, m248;
131   int m249, m250;
132   int m251, m252;
133   int m253, m254;
134   int m255, m256;
135   int m257, m258;
136   int m259, m260;
137   int m261, m262;
138   int m263, m264;
139   int m265, m266;
140   int m267, m268;
141   int m269, m270;
142   int m271, m272;
143   int m273, m274;
144   int m275, m276;
145   int m277, m278;
146   int m279, m280;
147   int m281, m282;
148   int m283, m284;
149   int m285, m286;
150   int m287, m288;
151   int m289, m290;
152   int m291, m292;
153   int m293, m294;
154   int m295, m296;
155   int m297, m298;
156   int m299, m300;
157   int m301, m302;
158   int m303, m304;
159   int m305, m306;
160   int m307, m308;
161   int m309, m310;
162   int m311, m312;
163   int m313, m314;
164   int m315, m316;
165   int m317, m318;
166   int m319, m320;
167   int m321, m322;
168   int m323, m324;
169   int m325, m326;
170   int m327, m328;
171   int m329, m330;
172   int m331, m332;
173   int m333, m334;
174   int m335, m336;
175   int m337, m338;
176   int m339, m340;
177   int m341, m342;
178   int m343, m344;
179   int m345, m346;
180   int m347, m348;
181   int m349, m350;
182   int m351, m352;
183   int m353, m354;
184   int m355, m356;
185   int m357, m358;
186   int m359, m360;
187   int m361, m362;
188   int m363, m364;
189   int m365, m366;
190   int m367, m368;
191   int m369, m370;
192   int m371, m372;
193   int m373, m374;
194   int m375, m376;
195   int m377, m378;
196   int m379, m380;
197   int m381, m382;
198   int m383, m384;
199   int m385, m386;
200   int m387, m388;
201   int m389, m390;
202   int m391, m392;
203   int m393, m394;
204   int m395, m396;
205   int m397, m398;
206   int m399, m400;
207   int m401, m402;
208   int m403, m404;
209   int m405, m406;
210   int m407, m408;
211   int m409, m410;
212   int m411, m412;
213   int m413, m414;
214   int m415, m416;
215   int m417, m418;
216   int m419, m420;
217   int m421, m422;
218   int m423, m424;
219   int m425, m426;
220   int m427, m428;
221   int m429, m430;
222   int m431, m432;
223   int m433, m434;
224   int m435, m436;
225   int m437, m438;
226   int m439, m440;
227   int m441, m442;
228   int m443, m444;
229   int m445, m446;
230   int m447, m448;
231   int m449, m450;
232   int m451, m452;
233   int m453, m454;
234   int m455, m456;
235   int m457, m458;
236   int m459, m460;
237   int m461, m462;
238   int m463, m464;
239   int m465, m466;
240   int m467, m468;
241   int m469, m470;
242   int m471, m472;
243   int m473, m474;
244   int m475, m476;
245   int m477, m478;
246   int m479, m480;
247   int m481, m482;
248   int m483, m484;
249   int m485, m486;
250   int m487, m488;
251   int m489, m490;
252   int m491, m492;
253   int m493, m494;
254   int m495, m496;
255   int m497, m498;
256   int m499, m500;
257   int m501, m502;
258   int m503, m504;
259   int m505, m506;
260   int m507, m508;
261   int m509, m510;
262   int m511, m512;
263   int m513, m514;
264   int m515, m516;
265   int m517, m518;
266   int m519, m520;
267   int m521, m522;
268   int m523, m524;
269   int m525, m526;
270   int m527, m528;
271   int m529, m530;
272   int m531, m532;
273   int m533, m534;
274   int m535, m536;
275   int m537, m538;
276   int m539, m540;
277   int m541, m542;
278   int m543, m544;
279   int m545, m546;
280   int m547, m548;
281   int m549, m550;
282   int m551, m552;
283   int m553, m554;
284   int m555, m556;
285   int m557, m558;
286   int m559, m560;
287   int m561, m562;
288   int m563, m564;
289   int m565, m566;
290   int m567, m568;
291   int m569, m570;
292   int m571, m572;
293   int m573, m574;
294   int m575, m576;
295   int m577, m578;
296   int m579, m580;
297   int m581, m582;
298   int m583, m584;
299   int m585, m586;
300   int m587, m588;
301   int m589, m590;
302   int m591, m592;
303   int m593, m594;
304   int m595, m596;
305   int m597, m598;
306   int m599, m600;
307   int m601, m602;
308   int m603, m604;
309   int m605, m606;
310   int m607, m608;
311   int m609, m610;
312   int m611, m612;
313   int m613, m614;
314   int m615, m616;
315   int m617, m618;
316   int m619, m620;
317   int m621, m622;
318   int m623, m624;
319   int m625, m626;
320   int m627, m628;
321   int m629, m630;
322   int m631, m632;
323   int m633, m634;
324   int m635, m636;
325   int m637, m638;
326   int m639, m640;
327   int m641, m642;
328   int m643, m644;
329   int m645, m646;
330   int m647, m648;
331   int m649, m650;
332   int m651, m652;
333   int m653, m654;
334   int m655, m656;
335   int m657, m658;
336   int m659, m660;
337   int m661, m662;
338   int m663, m664;
339   int m665, m666;
340   int m667, m668;
341   int m669, m670;
342   int m671, m672;
343   int m673, m674;
344   int m675, m676;
345   int m677, m678;
346   int m679, m680;
347   int m681, m682;
348   int m683, m684;
349   int m685, m686;
350   int m687, m688;
351   int m689, m690;
352   int m691, m692;
353   int m693, m694;
354   int m695, m696;
355   int m697, m698;
356   int m699, m700;
357   int m701, m702;
358   int m703, m704;
359   int m705, m706;
360   int m707, m708;
361   int m709, m710;
362   int m711, m712;
363   int m713, m714;
364   int m715, m716;
365   int m717, m718;
366   int m719, m720;
367   int m721, m722;
368   int m723, m724;
369   int m725, m726;
370   int m727, m728;
371   int m729, m730;
372   int m731, m732;
373   int m733, m734;
374   int m735, m736;
375   int m737, m738;
376   int m739, m740;
377   int m741, m742;
378   int m743, m744;
379   int m745, m746;
380   int m747, m748;
381   int m749, m750;
382   int m751, m752;
383   int m753, m754;
384   int m755, m756;
385   int m757, m758;
386   int m759, m760;
387   int m761, m762;
388   int m763, m764;
389   int m765, m766;
390   int m767, m768;
391   int m769, m770;
392   int m771, m772;
393   int m773, m774;
394   int m775, m776;
395   int m777, m778;
396   int m779, m780;
397   int m781, m782;
398   int m783, m784;
399   int m785, m786;
400   int m787, m788;
401   int m789, m790;
402   int m791, m792;
403   int m793, m794;
404   int m795, m796;
405   int m797, m798;
406   int m799, m800;
407   int m801, m802;
408   int m803, m804;
409   int m805, m806;
410   int m807, m808;
411   int m809, m810;
412   int m811, m812;
413   int m813, m814;
414   int m815, m816;
415   int m817, m818;
416   int m819, m820;
417   int m821, m822;
418   int m823, m824;
419   int m825, m826;
420   int m827, m828;
421   int m829, m830;
422   int m831, m832;
423   int m833, m834;
424   int m835, m836;
425   int m837, m838;
426   int m839, m840;
427   int m841, m842;
428   int m843, m844;
429   int m845, m846;
430   int m847, m848;
431   int m849, m850;
432   int m851, m852;
433   int m853, m854;
434   int m855, m856;
435   int m857, m858;
436   int m859, m860;
437   int m861, m862;
438   int m863, m864;
439   int m865, m866;
440   int m867, m868;
441   int m869, m870;
442   int m871, m872;
443   int m873, m874;
444   int m875, m876;
445   int m877, m878;
446   int m879, m880;
447   int m881, m882;
448   int m883, m884;
449   int m885, m886;
450   int m887, m888;
451   int m889, m890;
452   int m891, m892;
453   int m893, m894;
454   int m895, m896;
455   int m897, m898;
456   int m899, m900;
457   int m901, m902;
458   int m903, m904;
459   int m905, m906;
460   int m907, m908;
461   int m909, m910;
462   int m911, m912;
463   int m913, m914;
464   int m915, m916;
465   int m917, m918;
466   int m919, m920;
467   int m921, m922;
468   int m923, m924;
469   int m925, m926;
470   int m927, m928;
471   int m929, m930;
472   int m931, m932;
473   int m933, m934;
474   int m935, m936;
475   int m937, m938;
476   int m939, m940;
477   int m941, m942;
478   int m943, m944;
479   int m945, m946;
480   int m947, m948;
481   int m949, m950;
482   int m951, m952;
483   int m953, m954;
484   int m955, m956;
485   int m957, m958;
486   int m959, m960;
487   int m961, m962;
488   int m963, m964;
489   int m965, m966;
490   int m967, m968;
491   int m969, m970;
492   int m971, m972;
493   int m973, m974;
494   int m975, m976;
495   int m977, m978;
496   int m979, m980;
497   int m981, m982;
498   int m983, m984;
499   int m985, m986;
500   int m987, m988;
501   int m989, m990;
502   int m991, m992;
503   int m993, m994;
504   int m995, m996;
505   int m997, m998;
506   int m999, m1000;
507   int m1001, m1002;
508   int m1003, m1004;
509   int m1005, m1006;
510   int m1007, m1008;
511   int m1009, m1010;
512   int m1011, m1012;
513   int m1013, m1014;
514   int m1015, m1016;
515   int m1017, m1018;
516   int m1019, m1020;
517   int m1021, m1022;
518   int m1023, m1024;
519   int m1025, m1026;
520   int m1027, m1028;
521   int m1029, m1030;
522   int m1031, m1032;
523   int m1033, m1034;
524   int m1035, m1036;
525   int m1037, m1038;
526   int m1039, m1040;
527   int m1041, m1042;
528   int m1043, m1044;
529   int m1045, m1046;
530   int m1047, m1048;
531   int m1049, m1050;
532   int m1051, m1052;
533   int m1053, m1054;
534   int m1055, m1056;
535   int m1057, m1058;
536   int m1059, m1060;
537   int m1061, m1062;
538   int m1063, m1064;
539   int m1065, m1066;
540   int m1067, m1068;
541   int m1069, m1070;
542   int m1071, m1072;
543   int m1073, m1074;
544   int m1075, m1076;
545   int m1077, m1078;
546   int m1079, m1080;
547   int m1081, m1082;
548   int m1083, m1084;
549   int m1085, m1086;
550   int m1087, m1088;
551   int m1089, m1090;
552   int m1091, m1092;
553   int m1093, m1094;
554   int m1095, m1096;
555   int m1097, m1098;
556   int m1099, m1100;
557   int m1101, m1102;
558   int m1103, m1104;
559   int m1105, m1106;
560   int m1107, m1108;
561   int m1109, m1110;
562   int m1111, m1112;
563   int m1113, m1114;
564   int m1115, m1116;
565   int m1117, m1118;
566   int m1119, m1120;
567   int m1121, m1122;
568   int m1123, m1124;
569   int m1125, m1126;
570   int m1127, m1128;
571   int m1129, m1130;
572   int m1131, m1132;
573   int m1133, m1134;
574   int m1135, m1136;
575   int m1137, m1138;
576   int m1139, m1140;
577   int m1141, m1142;
578   int m1143, m1144;
579   int m1145, m1146;
580   int m1147, m1148;
581   int m1149, m1150;
582   int m1151, m1152;
583   int m1153, m1154;
584   int m1155, m1156;
585   int m1157, m1158;
586   int m1159, m1160;
587   int m1161, m1162;
588   int m1163, m1164;
589   int m1165, m1166;
590   int m1167, m1168;
591   int m1169, m1170;
592   int m1171, m1172;
593   int m1173, m1174;
594   int m1175, m1176;
595   int m1177, m1178;
596   int m1179, m1180;
597   int m1181, m1182;
598   int m1183, m1184;
599   int m1185, m1186;
600   int m1187, m1188;
601   int m1189, m1190;
602   int m1191, m1192;
603   int m1193, m1194;
604   int m1195, m1196;
605   int m1197, m1198;
606   int m1199, m1200;
607   int m1201, m1202;
608   int m1203, m1204;
609   int m1205, m1206;
610   int m1207, m1208;
611   int m1209, m1210;
612   int m1211, m1212;
613   int m1213, m1214;
614   int m1215, m1216;
615   int m1217, m1218;
616   int m1219, m1220;
617   int m1221, m1222;
618   int m1223, m1224;
619   int m1225, m1226;
620   int m1227, m1228;
621   int m1229, m1230;
622   int m1231, m1232;
623   int m1233, m1234;
624   int m1235, m1236;
625   int m1237, m1238;
626   int m1239, m1240;
627   int m1241, m1242;
628   int m1243, m1244;
629   int m1245, m1246;
630   int m1247, m1248;
631   int m1249, m1250;
632   int m1251, m1252;
633   int m1253, m1254;
634   int m1255, m1256;
635   int m1257, m1258;
636   int m1259, m1260;
637   int m1261, m1262;
638   int m1263, m1264;
639   int m1265, m1266;
640   int m1267, m1268;
641   int m1269, m1270;
642   int m1271, m1272;
643   int m1273, m1274;
644   int m1275, m1276;
645   int m1277, m1278;
646   int m1279, m1280;
647   int m1281, m1282;
648   int m1283, m1284;
649   int m1285, m1286;
650   int m1287, m1288;
651   int m1289, m1290;
652   int m1291, m1292;
653   int m1293, m1294;
654   int m1295, m1296;
655   int m1297, m1298;
656   int m1299, m1300;
657   int m1301, m1302;
658   int m1303, m1304;
659   int m1305, m1306;
660   int m1307, m1308;
661   int m1309, m1310;
662   int m1311, m1312;
663   int m1313, m1314;
664   int m1315, m1316;
665   int m1317, m1318;
666   int m1319, m1320;
667   int m1321, m1322;
668   int m1323, m1324;
669   int m1325, m1326;
670   int m1327, m1328;
671   int m1329, m1330;
672   int m1331, m1332;
673   int m1333, m1334;
674   int m1335, m1336;
675   int m1337, m1338;
676   int m1339, m1340;
677   int m1341, m1342;
678   int m1343, m1344;
679   int m1345, m1346;
680   int m1347, m1348;
681   int m1349, m1350;
682   int m1351, m1352;
683   int m1353, m1354;
684   int m1355, m1356;
685   int m1357, m1358;
686   int m1359, m1360;
687   int m1361, m1362;
688   int m1363, m1364;
689   int m1365, m1366;
690   int m1367, m1368;
691   int m1369, m1370;
692   int m1371, m1372;
693   int m1373, m1374;
694   int m1375, m1376;
695   int m1377, m1378;
696   int m1379, m1380;
697   int m1381, m1382;
698   int m1383, m1384;
699   int m1385, m1386;
700   int m1387, m1388;
701   int m1389, m1390;
702   int m1391, m1392;
703   int m1393, m1394;
704   int m1395, m1396;
705   int m1397, m1398;
706   int m1399, m1400;
707   int m1401, m1402;
708   int m1403, m1404;
709   int m1405, m1406;
710   int m1407, m1408;
711   int m1409, m1410;
712   int m1411, m1412;
713   int m1413, m1414;
714   int m1415, m1416;
715   int m1417, m1418;
716   int m1419, m1420;
717   int m1421, m1422;
718   int m1423, m1424;
719   int m1425, m1426;
720   int m1427, m1428;
721   int m1429, m1430;
722   int m1431, m1432;
723   int m1433, m1434;
724   int m1435, m1436;
725   int m1437, m1438;
726   int m1439, m1440;
727   int m1441, m1442;
728   int m1443, m1444;
729   int m1445, m1446;
730   int m1447, m1448;
731   int m1449, m1450;
732   int m1451, m1452;
733   int m1453, m1454;
734   int m1455, m1456;
735   int m1457, m1458;
736   int m1459, m1460;
737   int m1461, m1462;
738   int m1463, m1464;
739   int m1465, m1466;
740   int m1467, m1468;
741   int m1469, m1470;
742   int m1471, m1472;
743   int m1473, m1474;
744   int m1475, m1476;
745   int m1477, m1478;
746   int m1479, m1480;
747   int m1481, m1482;
748   int m1483, m1484;
749   int m1485, m1486;
750   int m1487, m14
```