National University of Science and Technology, NUST
School of Mechanical And Manufacturing Engineering, SMME
Department of Mechanical Engineering, DME

Prepared By:

Name	CMS ID	Class	Lab Manual
Muhammad Asim Shah	**470574**	**ME-15 "C"**	**07**

Lab Manual # 07 Lab Task

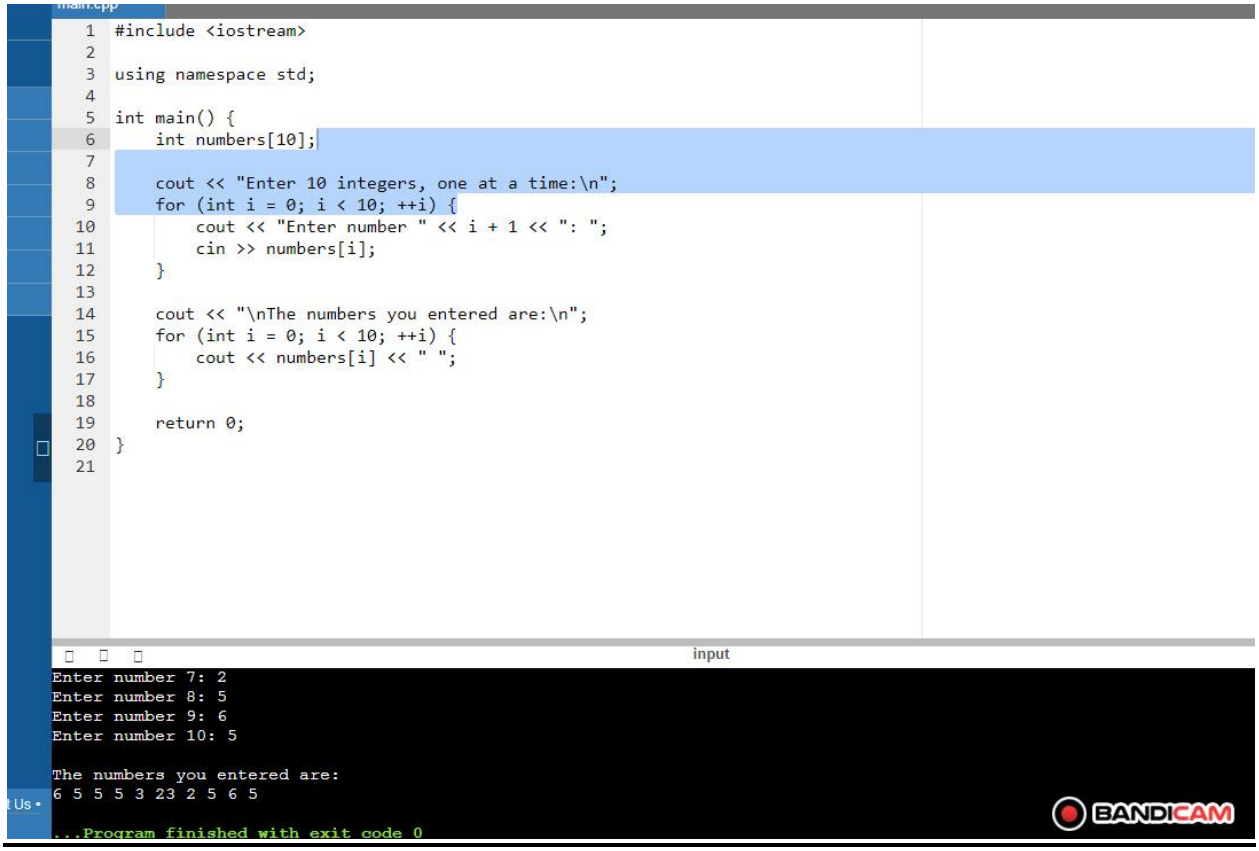
Task # 01: Take 10 integer inputs from user and store them in an array and print them on screen.

Input:

```
#include <iostream>

using namespace std;

int main() {

   int numbers[10];

cout << "Enter 10 integers, one at a time:\n";

   for (int i = 0; i < 10; ++i) {

      cout << "Enter number " << i + 1 << ": ";

      cin >> numbers[i];

   }
```

```
    cout << "\nThe numbers you entered are:\n";

    for (int i = 0; i < 10; ++i) {

        cout << numbers[i] << " ";

    }


    return 0;

}
```

Output:



TASK 2: Write a program to find the sum and product of all elements of an array with 5 integer elements.

Input:

```
#include <iostream>

using namespace std;

int main() {

   const int size = 5;

   int diamond[size][size];

   for (int i = 0; i < size; ++i) {

      for (int j = 0; j < size; ++j) {

         diamond[i][j] = 0;

      }

   }

   for (int i = 0; i < size / 2 + 1; ++i) {

      diamond[i][size / 2 - i] = 1;

      diamond[i][size / 2 + i] = 1;

   }

for (int i = size / 2 + 1; i < size; ++i) {

      diamond[i][i - size / 2] = 1;

      diamond[i][size - (i - size / 2) - 1] = 1;

   }

for (int i = 0; i < size; ++i) {

      for (int j = 0; j < size; ++j) {

         if (diamond[i][j] == 1)

            cout << "* ";
```

```
        else

            cout << "  ";

        }

        cout << endl;

    }

return 0;

}
```

Output:

```
 6      int numbers[5];
 7
 8      cout << "Enter 5 integers, one at a time:\n";
 9      for (int i = 0; i < 5; ++i) {
10          cout << "Enter number " << i + 1 << ": ";
11          cin >> numbers[i];
12      }
13
14      int sum = 0;
15      int product = 1;
16
17      // Calculate the sum and product of the elements
18      for (int i = 0; i < 5; ++i) {
19          sum += numbers[i];
20          product *= numbers[i];
21      }
22
23      cout << "\nThe sum of the numbers is: " << sum << endl;
24      cout << "The product of the numbers is: " << product << endl;
```

input

```
Enter 5 integers, one at a time:
Enter number 1: 4
Enter number 2: 3
Enter number 3: 2
Enter number 4: 7
Enter number 5: 9

The sum of the numbers is: 25
The product of the numbers is: 1512


...Program finished with exit code 0
Press ENTER to exit console.
```

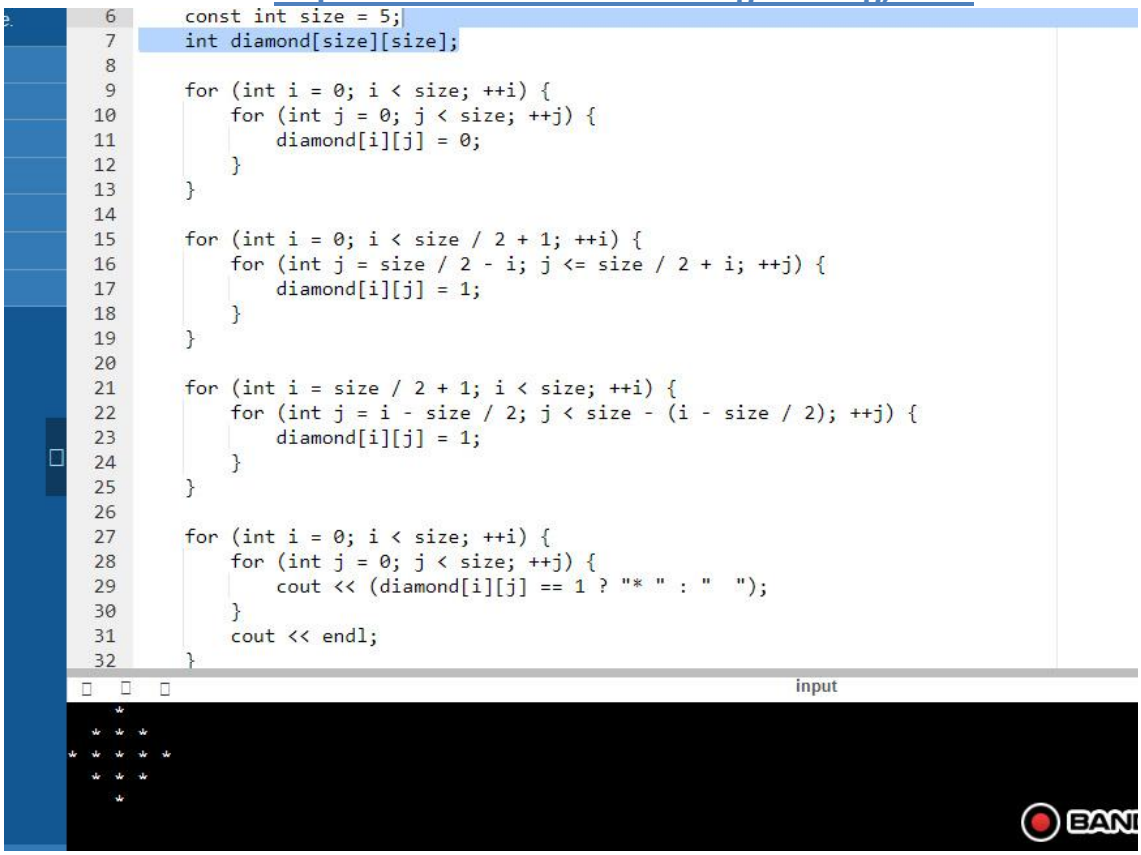
TASK 3: Print diamond pattern using a single array.

Input:

```
#include <iostream>


using namespace std;


int main() {
   const int size = 5;
   int diamond[size][size];


   for (int i = 0; i < size; ++i) {
      for (int j = 0; j < size; ++j) {
         diamond[i][j] = 0;
      }
   }


   for (int i = 0; i < size / 2 + 1; ++i) {
      for (int j = size / 2 - i; j <= size / 2 + i; ++j) {
         diamond[i][j] = 1;
      }
   }
```

```
    for (int i = size / 2 + 1; i < size; ++i) {

        for (int j = i - size / 2; j < size - (i - size / 2); ++j) {

            diamond[i][j] = 1;

        }

    }


    for (int i = 0; i < size; ++i) {

        for (int j = 0; j < size; ++j) {

            cout << (diamond[i][j] == 1 ? "* " : "  ");

        }

        cout << endl;

    }


    return 0;

}
```

Output:



```
6    const int size = 5;
7    int diamond[size][size];
8
9    for (int i = 0; i < size; ++i) {
10       for (int j = 0; j < size; ++j) {
11           diamond[i][j] = 0;
12       }
13   }
14
15   for (int i = 0; i < size / 2 + 1; ++i) {
16       for (int j = size / 2 - i; j <= size / 2 + i; ++j) {
17           diamond[i][j] = 1;
18       }
19   }
20
21   for (int i = size / 2 + 1; i < size; ++i) {
22       for (int j = i - size / 2; j < size - (i - size / 2); ++j) {
23           diamond[i][j] = 1;
24       }
25   }
26
27   for (int i = 0; i < size; ++i) {
28       for (int j = 0; j < size; ++j) {
29           cout << (diamond[i][j] == 1 ? "* " : "  ");
30       }
31       cout << endl;
32   }
```

input

```
    *
  * * *
* * * * *
  * * *
    *
```