Assignment 2

C++ Program for

1. To Print factorial of the number
2. Sum of array of 10 sum of odd no. and even no.
3. Matrix addition of 3x3
4. Size of all data types

1.

Ans.

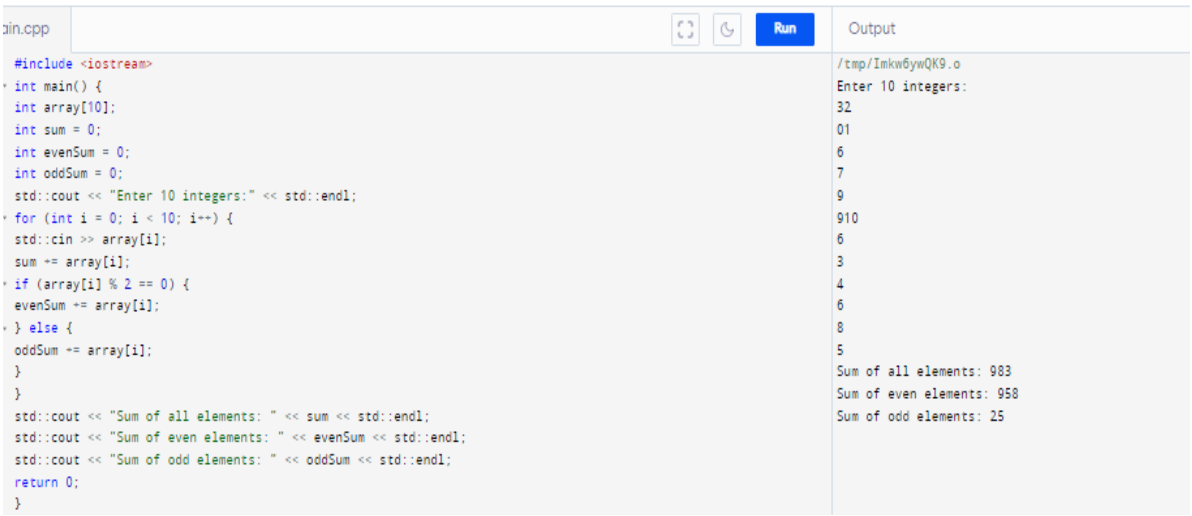
```
#include <iostream>
unsigned long long factorial(int n) {
if (n == 0 || n == 1)
return 1;
else
return n * factorial(n - 1);
}
int main() {
int num;
std::cout << "Enter a number: ";
std::cin >> num;
if (num < 0) {
std::cout << "Please enter a non-negative number." << std::endl;
} else {
std::cout << "Factorial of " << num << " = " << factorial(num) << std::endl;
}
return 0;
}
```

Output :

2.
Ans.

```
#include <iostream>
int main() {
int array[10];
int sum = 0;
int evenSum = 0;
int oddSum = 0;
std::cout << "Enter 10 integers:" << std::endl;
for (int i = 0; i < 10; i++) {
std::cin >> array[i];
sum += array[i];
if (array[i] % 2 == 0) {
evenSum += array[i];
} else {
oddSum += array[i];
}
}
std::cout << "Sum of all elements: " << sum << std::endl;
std::cout << "Sum of even elements: " << evenSum << std::endl;
std::cout << "Sum of odd elements: " << oddSum << std::endl;
return 0;
}
```

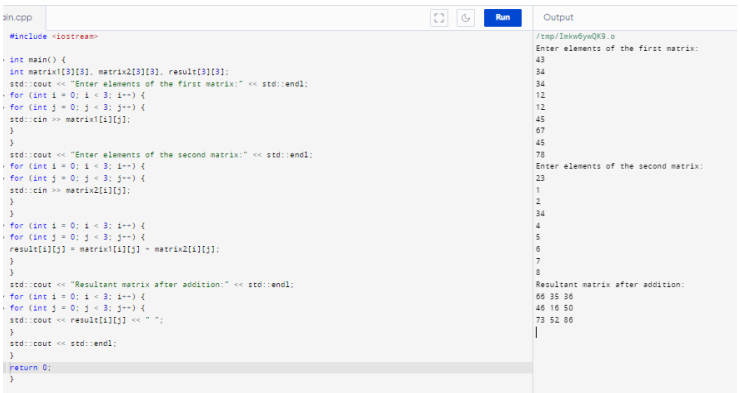
Output :



3.
Ans.

```
#include <iostream>
int main() {
int matrix1[3][3], matrix2[3][3], result[3][3];
std::cout << "Enter elements of the first matrix:" << std::endl;
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
std::cin >> matrix1[i][j];
}
}
std::cout << "Enter elements of the second matrix:" << std::endl;
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
std::cin >> matrix2[i][j];
}
}
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
result[i][j] = matrix1[i][j] + matrix2[i][j];
}
}
std::cout << "Resultant matrix after addition:" << std::endl;
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
std::cout << result[i][j] << " ";
}
std::cout << std::endl;
}
return 0;
}
```

Output :



4.
Ans.

```
#include <iostream>

enum Color { RED, GREEN, BLUE };
struct SampleStruct { int x; double y; };

int main() {
    std::cout << "Size of char: " << sizeof(char) << " bytes\n";
    std::cout << "Size of short: " << sizeof(short) << " bytes\n";
    std::cout << "Size of int: " << sizeof(int) << " bytes\n";
    std::cout << "Size of long: " << sizeof(long) << " bytes\n";
    std::cout << "Size of long long: " << sizeof(long long) << " bytes\n";
    std::cout << "Size of float: " << sizeof(float) << " bytes\n";
    std::cout << "Size of double: " << sizeof(double) << " bytes\n";
    std::cout << "Size of long double: " << sizeof(long double) << " bytes\n";
    std::cout << "Size of unsigned char: " << sizeof(unsigned char) << " bytes\n";
    std::cout << "Size of unsigned short: " << sizeof(unsigned short) << " bytes\n";
    std::cout << "Size of unsigned int: " << sizeof(unsigned int) << " bytes\n";
    std::cout << "Size of unsigned long: " << sizeof(unsigned long) << " bytes\n";
    std::cout << "Size of unsigned long long: " << sizeof(unsigned long long) << " bytes\n";
    std::cout << "Size of char*: " << sizeof(char *) << " bytes\n";
    std::cout << "Size of int*: " << sizeof(int *) << " bytes\n";
    std::cout << "Size of float*: " << sizeof(float *) << " bytes\n";
    std::cout << "Size of double*: " << sizeof(double *) << " bytes\n";
    std::cout << "Size of void*: " << sizeof(void *) << " bytes\n";
    std::cout << "Size of enum Color: " << sizeof(Color) << " bytes\n";
    std::cout << "Size of struct SampleStruct: " << sizeof(SampleStruct) << " bytes\n";

    return 0;
}
```

Output :