

Q1:

```
#include <iostream>

using namespace std;

int main() {

    int age;

    string name;

    cout<<"enter the name : "<<endl;

    cin>>name;

    cout<<"enter the age : "<<endl;

    cin>>age;

    cout<<"hello" <<name <<"you are" <<age <<"years old";

    return 0;

}
```

enter the name :

uio

enter the age :

78

hellouiouyou are78years old

Q2

```
int a,b;

    cout<<"enter 2 values:";

    cin>>a>>b;

    cout<<"sum of no.s is "<<a+b;
```

enter 2 values:5 6

sum of no.s is 11

Q3:

```
int a, b ;

    char opr;

    cout<<"enter 2 values:";

    cin>>a>>b;

    cout<<"enter operator:";
```

```
cin>>opr;

switch(opr){

    case'+':

        cout<<a+b;

        break;

    case'-':

        cout<<a-b;

        break;

    case'*':

        cout<<a*b;

        break;

    case'/':

        cout<<a/b;

        break;

    default:"no case";

}

return 0;

}
```

enter 2 values:7 9

enter operator:*

63

Q4:

```
float a,b ;
```

```
    cout<<"enter 2 values:";
```

```
    cin>>a>>b;
```

```
    float c=a/b;
```

```
    cout<<fixed<<setprecision(2)<<c;
```

enter 2 values:8 9

0.89

Q5:#include <iostream>

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    double p,r,n,t,ci,a;
```

```
    cout<<"enter the principle:";
```

```
    cin>>p;
```

```
    cout<<"enter the rate:";
```

```
    cin>>r;
```

```
    cout<<"enter the number of time :";
```

```
    cin>>n;
```

```
    cout<<"enter the time:";
```

```
    cin>>t;
```

```
    r=r/100;
```

```
    a=p*pow((1+r/n),(n*t));
```

```
    ci=a-p;
```

```
    cout<<"the amount:"<<a<<endl;
```

```
    cout<<"the compound intrest:"<<ci<<endl;
```

```
    return 0;
```

```
}
```

enter the principle:1000

enter the rate:5

enter the number of time :4

enter the time:60

the amount:19715.5

the compound intrest:18715.5

Q6:

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    float l,b,a,p;
```

```
    cout<<"enter length:";
```

```
    cin>>l;
```

```
    cout<<"enter the breadth:";
```

```
    cin>>b;
```

```
    a=l*b;
```

```
    cout<<"area:"<<a<<endl;
```

```
    p=2*(l+b);
```

```
    cout<<"perimeter:"<<p<<endl;
```

```
    return 0;
```

```
}
```

enter length:5.3

enter the breadth:2

area:10.6

perimeter:14.6

Q7:

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {  
  
    int a,b,c;  
  
    cout<<"enter 2 values:";  
  
    cin>>a>>b;  
  
    c=a;  
  
    a=b;  
  
    b=c;  
  
    cout<<a<<endl;  
  
    cout<<b<<endl;  
  
    return 0;  
  
}
```

enter 2 values:5 6

6

5

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {  
  
    int a,b,c;  
  
    cout<<"enter 2 values:";  
  
    cin>>a>>b;  
  
    a=a+b;  
  
    b=a-b;  
  
    a=a-b;  
  
    cout<<a<<endl;  
  
    cout<<b<<endl;  
  
    return 0;  
  
}
```

enter 2 values:5 6

6

5

Q8-

```
#include <iostream>

using namespace std;

int main() {

    int age;

    string name;

    cout<<"enter the name:";

    getline(cin,name);

    cout<<"enter the age:";

    cin>>age;

    cout<<"hello"<<name<<"i have lived for "<<age;


    return 0;

}
```

enter the name:hui

enter the age:7

hellohuii have lived for 7

Q9#include <iostream>

#include<iomanip>

using namespace std;

```
int main() {

    string ch;

    int num;

    float numm;

    cout<<"enter the ch:";
```

```

getline(cin,ch);

cout<<"enter the num:";

cin>>num;

cout<<"enter the floating value:";

cin>>numm;

cout<<" +-----+-----+-----+"<<endl;

cout<<"| character   | integer    | flaoting  |"<<endl;

cout<<" +-----+-----+-----+"<<endl;

cout<<setw(15)<<ch<<"|"<<setw(15)<<num<<"|"<<setw(15)<<fixed<<setprecision(2)<<numm<<"|"<<endl;

cout<<" +-----+-----+-----+"<<endl;


return 0;

}

```

enter the ch:poi

enter the num:896

enter the floating value:21.3

```

+-----+-----+-----+
| character   | integer    | flaoting  |
+-----+-----+-----+
      poi|      896|      21.30|
+-----+-----+-----+

```

Q10:#include <iostream>

using namespace std;

int main() {

int a,b;

cout<<"enter 2 values:";

cin>>a>>b;

if(b!=0 && a%b==0){

```

        cout<<b<<" is multiple of"<<a<<endl;

    }else{

        cout<<b<<" is not multiple of"<<a<<endl;

    }

    return 0;

}

```

enter 2 values:8 4

4 is multiple of8

```

Q11#include <iostream>

using namespace std;

int countbits(int num){

    int count=0;

    while(num>0){

        if(num &1){

            count++;

        }

        num=num>>1;

    }

    return count;

}

int main() {

    int num;

    cout<<"enter the integer :";

    cin>>num;

    unsigned int unsignednum=static_cast<unsigned int>(num);

    int result=countbits(unsignednum);

    cout<<"noo of 1 bits:"<<result<<endl;

```



```
    return 0;
}

enter the integer :7

noo of 1 bits:3
```

Q12

```
#include <iostream>

#include <cmath>

using namespace std;

int main() {

    int a, b, c;

    cout << "Enter coefficients a, b, and c: ";

    cin >> a >> b >> c;

)

    if (a == 0) {

        if (b == 0) {

            cout << "No solution exists (invalid equation)." << endl;

        } else {

            int root = -c / b;

            cout << "Linear equation solution: x = " << root << endl;

        }

    } else {

        int D = b * b - 4 * a * c;

        if (D > 0) {

            int sqrtD = sqrt(D); // sqrt(D) gives the square root
```

```

if (sqrtD * sqrtD == D) { // Check if sqrt(D) is an integer

    int root1 = (-b + sqrtD) / (2 * a);

    int root2 = (-b - sqrtD) / (2 * a);

    cout << "Two distinct real roots: x1 = " << root1 << ", x2 = " << root2 << endl;

} else {

    cout << "Non-perfect square discriminant. Cannot calculate integer roots." << endl;

}

} else if (D == 0) {

    // One real and equal root

    int root = -b / (2 * a);

    cout << "One real root (equal roots): x = " << root << endl;

} else {

    // Complex roots

    cout << "Complex roots: Cannot solve with integer calculations." << endl;

}

}

return 0;

}

```

Enter coefficients a, b, and c: 1 -5 6

Two distinct real roots: x1 = 3, x2 = 2

Q13:

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main() {
```

```
    int balance=5000;
```

```
    int pin=1234;
```

```
    int enteredpin;
```

```

int option;

int amount;

cout<<"enter the pin:";

cin>>enteredpin;

if(enteredpin != pin){

    cout<<"wrong pin.access denied";

    return 0;

}

do{

    cout<<"1.withdrwal";

    cout<<"2.deposit";

    cout<<"3.balance inquiry";

    cout<<"4.exit";

    cout<<"enter the option(1-4):";

    cin>>option;

    switch(option){

        case 1:

            cout<<"enter the amount to be withdrawn:";

            if(amount<=0){

                cout<<"it has to be grater than zero";

            }else if(amount >balance){

                cout<<"sorry the amount is not available";

            }else{

                balance-=amount;

                cout<<"the amount is withdrawn"<<balance<<endl;

            }

        }

    }

    break;

    case 2:

```

```
cout<<"enter the amount to deposit:";

cin>>amount;

if(amount<=0){

    cout<<"amount should be more than zero";

}else if{

    balance+=amount;

    cout<<"deposit successfully.new balance:"<<balance<<endl;

}

break;

case 3:

cout<<"your current balance:"<<balance<<endl;

break;

case 4:

cout<<"exit";

default:

cout<<"invalid option";

}while(option!=4)

}
```

```
return 0;

}
```

Enter your PIN: 1234

ATM Menu:

1. Withdraw
2. Deposit
3. Balance Inquiry

4. Exit

Choose an option (1-4): 1

Enter amount to withdraw: 1000

Withdrawal successful. New balance: 4000

Q14

```
#include <iostream>

#include<cmath>

using namespace std;

int main() {

    int s1,s2,s3;

    cout<<"enter 3 values:";

    cin>>s1>>s2>>s3;

    if(s1==s2&& s2==s3&& s1==s3){

        cout<<"it is an equilateral triangle:";

    }else if(s1==s2 | | s2==s3 | | s1==s3){

        cout<<"it is an isosceles triangle";

    }else{

        cout<<"it is an scalene triangle";

    }

    return 0;

}
```

enter 3 values:65 45 65

it is an isosceles triangle

Q15

```
#include <iostream>

#include<cmath>

using namespace std;

int main() {
```

```
int a ,b,c,d,e;

cout<<"enter 4 values:";

cin>>a>>b>>c>>d;

e=(a^b)+(c&d);

cout<<e<<endl;
```

```
return 0;

}

enter 4 values:5 3 7 2

8
```

Q16:

```
#include <iostream>

using namespace std;

bool isoppositesigns(int a,int b){

    return (a^b)<0;

}

int main() {

    int x=-2,y=10;

    if(isoppositesigns(x,y)){

        cout<<"are of opposite signs"<<endl;

    }else{

        cout<<"are of same sign"<<endl;

    }

    return 0;

}
```

Output:

are of opposite signs

q17:

```
#include <iostream>
```

```
#include <cmath> // For abs()
```

```
using namespace std;
```

```
int manualIntegerDivision(int dividend, int divisor) {
```

```
    if (divisor == 0) {
```

```
        throw runtime_error("Division by zero is not allowed.");
```

```
    }
```

```
    bool isNegative = (dividend < 0) ^ (divisor < 0); // Check if result should be negative
```

```
    dividend = abs(dividend);
```

```
    divisor = abs(divisor);
```

```
    int quotient = 0;
```

```
    while (dividend >= divisor) {
```

```
        dividend -= divisor;
```

```
        quotient++;
```

```
    }
```

```
    return isNegative ? -quotient : quotient;
```

```
}
```

```
double manualFloatingPointDivision(int dividend, int divisor, int precision = 6) {
```

```
    if (divisor == 0) {
```

```
        throw runtime_error("Division by zero is not allowed.");
```

```
    }
```

```
bool isNegative = (dividend < 0) ^ (divisor < 0);
```

```
dividend = abs(dividend);
```

```
divisor = abs(divisor);
```

```
int integerPart = manualIntegerDivision(dividend, divisor);
```

```
int remainder = dividend - (integerPart * divisor);
```

```
double result = integerPart;
```

```
double fraction = 0.0;
```

```
double factor = 0.1;
```

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

```
    remainder *= 10;
```

```
    int fractionalDigit = manualIntegerDivision(remainder, divisor);
```

```
    fraction += fractionalDigit * factor;
```

```
    remainder -= fractionalDigit * divisor;
```

```
    factor /= 10;
```

```
}
```

```
result += fraction;
```

```
return isNegative ? -result : result;
```

```
}
```

```
int main() {
```

```
    int num1, num2;
```

```
    cout << "Enter dividend and divisor: ";
```

```
    cin >> num1 >> num2;
```

```
    try {
```

```
        int intDiv = manualIntegerDivision(num1, num2);
```



```
double floatDiv = manualFloatingPointDivision(num1, num2);

cout << "Integer Division: " << intDiv << endl;

cout << "Floating Point Division: " << floatDiv << endl;

} catch (runtime_error &e) {

    cout << "Error: " << e.what() << endl;

}

return 0;

}
```

OUTPUT:

```
Output

Enter dividend and divisor: 10 3
Integer Division: 3
Floating Point Division: 3.33333

=== Code Execution Successful ===
```

Q19:

```
main.cpp  [ ]  [ ]  [ ]  Share  Run  Output

3  #include <iostream>
4  using namespace std;
5  int main() {
6      int marks;
7      cout << "Enter student's marks (0-100): ";
8      cin >> marks;
9      if (marks < 0 || marks > 100) {
10         cout << "Invalid marks! Please enter a value between 0 and 100.\n";
11     }
12     else {
13         // Determine grade
14         if (marks >= 90)
15             cout << "Grade: A\n";
16         else if (marks >= 80)
17             cout << "Grade: B\n";
18         else if (marks >= 70)
19             cout << "Grade: C\n";
20         else if (marks >= 60)
21             cout << "Grade: D\n";
22         else
23             cout << "Grade: F\n";
24     }
25
26     return 0;
27 }
```

Enter student's marks (0-100): 87
Grade: B

=== Code Execution Successful ===

Q29:

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int choice;
```

```
    double num1,num2,result;
```

```
    do{
```

```
        cout<<"1.add"<<endl;
```

```
        cout<<"2. sub"<<endl;
```

```
        cout<<"3.mult"<<endl;
```

```
        cout<<"4.div"<<endl;
```

```
        cout<<"5. exit"<<endl;
```

```
        cout<<"enter your choice:";
```

```
        cin>>choice;
```

```
        if(choice>=1 && choice<=4){
```

```
            cout<<"enter 2 numbers:"<<endl;
```

```
            cin>>num1>>num2;
```

```
        }
```

```

switch(choice){

    case 1:

        result=num1+num2;

        cout<<"result is:"<<result<<endl;

        break;

    case 2:

        result=num1-num2;

        cout<<"result is:"<<result<<endl;

        break;

    case 3:

        result=num1*num2;

        cout<<"result is:"<<result<<endl;

        break;

    case 4:

        if(num2!=0){

            result=num1/num2;

            cout<<"result is:"<<result<<endl;

        }else{

            cout<<"division is not allowed"<<endl;

        }

        break;

    case 5:

        cout<<"exiting the program "<<endl;

        default:

            cout<<"invalid choice";

        }

}while(choice!=5);

return 0;

}

```

Output:

Output

```
1.add
2. sub
3.mult
4.div
5. exit
enter your choice:3
enter 2 numbers:
5 6
result is:30
1.add
2. sub
3.mult
4.div
5. exit
enter your choice:5
exiting the program
invalid choice
```

```
=== Code Execution Successful ===
```

main.cpp	Output
<pre>1 // Online C++ compiler to run C++ program online 2 #include <iostream> 3 using namespace std; 4 int main() { 5 int year; 6 cout << "Enter a year: "; 7 cin >> year; 8 if (year % 4 == 0) { 9 if (year % 100 == 0) { 10 if (year % 400 == 0) { 11 cout << year << " is a Leap Year.\n"; 12 } else { 13 cout << year << " is NOT a Leap Year.\n"; 14 } 15 } else { 16 cout << year << " is a Leap Year.\n"; 17 } 18 } else { 19 cout << year << " is NOT a Leap Year.\n"; 20 } 21 22 return 0; 23 } 24</pre>	<p>Enter a year: 2023 2023 is NOT a Leap Year.</p> <p>=== Code Execution Successful ===</p>

Q22:

main.cpp	Output
<pre>1 // Online C++ compiler to run C++ program online 2 #include <iostream> 3 using namespace std; 4 5 int main() { 6 int n; 7 cout << "Enter the number of Fibonacci numbers to print: "; 8 cin >> n; 9 int first = 0, second = 1, next; 10 cout << "Fibonacci Series: "; 11 12 for (int i = 0; i < n; i++) { 13 cout << first << " "; 14 next = first + second; 15 first = second; 16 second = next; 17 } 18 19 cout << endl; 20 return 0; 21 } 22</pre>	<p>Enter the number of Fibonacci numbers to print: 10 Fibonacci Series: 0 1 1 2 3 5 8 13 21 34</p> <p>=== Code Execution Successful ===</p>

Q23:

main.cpp	Output
<pre>2 #include <iostream> 3 using namespace std; 4 int main() { 5 int num, i = 2; 6 bool isPrime = true; 7 cout << "Enter a number: "; 8 cin >> num; 9 if (num <= 1) { 10 isPrime = false; 11 } else { 12 while (i * i <= num) { // Checking up to sqrt(num) 13 if (num % i == 0) { 14 isPrime = false; 15 break; 16 } 17 i++; 18 } 19 } 20 21 if (isPrime) 22 cout << num << " is a prime number." << endl; 23 else 24 cout << num << " is not a prime number." << endl; 25 26 return 0; 27 }</pre>	<pre>Enter a number: 4 4 is not a prime number. === Code Execution Successful ===</pre>

Q24:

main.cpp	Output
<pre>1 2 #include <iostream> 3 using namespace std; 4 5 int main() { 6 int n; 7 long long factorial = 1; 8 cout << "Enter a number: "; 9 cin >> n; 10 11 int i = 1; 12 do { 13 factorial *= i; 14 i++; 15 } while (i <= n); 16 17 18 cout << "Factorial of " << n << " is: " << factorial << endl; 19 20 return 0; 21 } 22</pre>	<pre>Enter a number: 5 Factorial of 5 is: 120 === Code Execution Successful ===</pre>

Q26:

main.cpp	Output
<pre>1 // Online C++ compiler to run C++ program online 2 #include <iostream> 3 using namespace std; 4 bool isPrime(int num) { 5 if (num < 2) return false; 6 for (int i = 2; i * i <= num; i++) { 7 if (num % i == 0) return false; 8 } 9 return true; 10 } 11 int main() { 12 int n, count = 0, num = 2; 13 cout << "Enter the number of prime numbers to generate: "; 14 cin >> n; 15 while (count < n) { 16 if (isPrime(num)) { 17 cout << num << " "; 18 count++; 19 } 20 num++; 21 } 22 cout << endl; 23 return 0; 24 } 25</pre>	<pre>Enter the number of prime numbers to generate: 6 2 3 5 7 11 13 === Code Execution Successful ===</pre>

Q27:

main.cpp	Output
<pre>1 #include <iostream> 2 #include <cmath> 3 using namespace std; 4 int main() { 5 int start, end; 6 cout << "Enter the start of the range: "; 7 cin >> start; 8 cout << "Enter the end of the range: "; 9 cin >> end; 10 cout << "Armstrong numbers between " << start << " and " << end 11 << " are:\n"; 12 for (int num = start; num <= end; num++) { 13 int sum = 0, temp = num, digits = 0; 14 for (int tempNum = num; tempNum > 0; tempNum /= 10) { 15 digits++; 16 } 17 for (int tempNum = num; tempNum > 0; tempNum /= 10) { 18 int digit = tempNum % 10; 19 sum += pow(digit, digits); 20 } 21 if (sum == num) { 22 cout << num << " "; 23 } 24 } 25 cout << endl; 26 return 0; </pre>	<pre>Enter the start of the range: 1 Enter the end of the range: 500 Armstrong numbers between 1 and 500 are: 1 2 3 4 5 6 7 8 9 153 370 371 407 === Code Execution Successful ===</pre>

Q29:

main.cpp

Share

Run

```
1
2 #include <iostream>
3
4 int main() {
5     int number = 51;
6
7     while (number % 7 != 0) {
8         number++;
9     }
10
11     std::cout << "The first number greater than 50 that is divisible
        by 7 is: " << number << std::endl;
12
13     return 0;
14 }
15
```

Output

The first number greater than 50 that is divisible by 7 is: 56

=== Code Execution Successful ===

Q30:

main.cpp

Share

Run

```
1 // Online C++ compiler to run C++ program online
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int sum=0;
6     for (int i=1;i<=500;i++){
7         if(i%3==0 || i%7==0){
8             continue;
9         }if(sum +i>1000)
10             break;
11
12         cout<<i<<" ";
13         sum+=i;
14     }
15     cout<<"total sum"<<sum<<endl;
16
17     return 0;
18 }
```

Output

1 2 4 5 8 10 11 13 16 17 19 20 22 23 25 26 29 31 32 34
 37 38 40 41 43 44 46 47 50 52 53 55 58 total sum952

=== Code Execution Successful ===

Q31:

main.cpp	Output
<pre>1 #include <iostream> 2 using namespace std; 3 int reverseNumber(int num) { 4 int reversed = 0; 5 while (num > 0) { 6 reversed = reversed * 10 + num % 10; 7 num /= 10; 8 } 9 return reversed; 10 } 11 bool isPalindrome(int num) { 12 return num == reverseNumber(num); 13 } 14 int main() { 15 int num; 16 cout << "Enter a number: "; 17 cin >> num; 18 while (true) { 19 int reversed = reverseNumber(num); 20 cout << "Reversed Number: " << reversed << endl; 21 if (num == reversed) { 22 cout << "Palindrome detected: " << num << endl; 23 break; 24 } 25 num = reversed; 26 } 27 }</pre>	<pre>Enter a number: 141 Reversed Number: 141 Palindrome detected: 141 === Code Execution Successful ===</pre>

Q35:

main.cpp	Output
<pre>1 #include <iostream> 2 using namespace std; 3 int gcd(int a, int b) { 4 while (b != 0) { 5 int temp = b; 6 b = a % b; 7 a = temp; 8 } 9 return a; 10 } 11 int lcm(int a, int b) { 12 return (a * b) / gcd(a, b); 13 } 14 15 int main() { 16 int num1, num2; 17 cout << "Enter two numbers: "; 18 cin >> num1 >> num2; 19 cout << "GCD of " << num1 << " and " << num2 << " is: " << gcd 20 (num1, num2) << endl; 21 cout << "LCM of " << num1 << " and " << num2 << " is: " << lcm 22 (num1, num2) << endl; 23 return 0; 24 }</pre>	<pre>Enter two numbers: 12 18 GCD of 12 and 18 is: 6 LCM of 12 and 18 is: 36 === Code Execution Successful ===</pre>

Q32

```
#include <iostream>
```

```
#include <limits.h>
```

```
using namespace std;
```

```
int findSecondLargest(int arr[], int size) {
```

```
    if (size < 2) {
```

```
        cout << "Array should have at least two elements.\n";
```

```
        return -1;
```

```
    }
```

```
    int first = INT_MIN, second = INT_MIN;
```

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

```
        if (arr[i] > first) {
```

```
            second = first;
```

```
            first = arr[i];
```

```
        } else if (arr[i] > second && arr[i] != first) {
```

```
            second = arr[i];
```

```
        }
```

```
    }
```

```
    return (second == INT_MIN) ? -1 : second;
```

```
}
```

```
int main() {
```

```
    int size;
```

```
    cout << "Enter the number of elements: ";
```

```
    cin >> size;
```

```
    int arr[size];
```

```

cout << "Enter the elements: ";
for (int i = 0; i < size; i++) {
    cin >> arr[i];
}

int secondLargest = findSecondLargest(arr, size);
if (secondLargest != -1)
    cout << "Second largest element: " << secondLargest << endl;
else
    cout << "No second largest element found.\n";

return 0;
}

```

Output:

Enter the number of elements: 5

Enter the elements: 10 20 5 30 25

Q33

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```

bool canBeRepresentedExactly(double num) {
    int exponent;
    double mantissa = frexp(num, &exponent);

    while (mantissa != floor(mantissa)) {
        mantissa *= 2;
        exponent--;
    }
}

```

```

        return exponent >= -52; // IEEE 754 double precision has 52-bit mantissa
    }

int main() {
    double num;
    cout << "Enter a floating-point number: ";
    cin >> num;

    if (canBeRepresentedExactly(num)) {
        cout << num << " can be represented exactly in binary.\n";
    } else {
        cout << num << " cannot be represented exactly in binary.\n";
    }

    return 0;
}

```

Output:

Enter a floating-point number: 0.5

0.5 can be represented exactly in binary.

Q34:

```

#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int rows, cols;
    cout << "Enter the number of rows and columns: ";
    cin >> rows >> cols;

    int arr[rows][cols];

```

```

cout << "Enter the elements of the matrix:\n";

for (int i = 0; i < rows; i++)
    for (int j = 0; j < cols; j++)
        cin >> arr[i][j];

cout << "Formatted 2D Array:\n";
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++)
        cout << setw(5) << arr[i][j] << " "; // Ensuring proper alignment
    cout << endl;
}

return 0;
}

```

Output:

Enter the number of rows and columns: 3 3

Enter the elements of the matrix:

1 22 333

4444 55 6

7 888 9999

Formatted 2D Array:

```

1  22 333
4444 55  6
7 888 9999

```

Q18

```
#include <iostream>
```

```
using namespace std;
```

```
// Function to perform circular left shift
```

```

unsigned int circularLeftShift(unsigned int num, int shift, int bitSize = 8) {
    shift %= bitSize; // Ensure shift is within range

    return ((num << shift) | (num >> (bitSize - shift))) & ((1 << bitSize) - 1);
}

// Function to perform circular right shift
unsigned int circularRightShift(unsigned int num, int shift, int bitSize = 8) {
    shift %= bitSize; // Ensure shift is within range

    return ((num >> shift) | (num << (bitSize - shift))) & ((1 << bitSize) - 1);
}

int main() {
    unsigned int num;

    int shift, bitSize = 8; // Default bit size set to 8 bits

    // Get user input
    cout << "Enter an integer: ";
    cin >> num;

    cout << "Enter shift value: ";
    cin >> shift;

    // Perform circular shifts
    unsigned int leftShifted = circularLeftShift(num, shift, bitSize);
    unsigned int rightShifted = circularRightShift(num, shift, bitSize);

    // Display results
    cout << "Original Number (8-bit): " << (num & ((1 << bitSize) - 1)) << endl;
    cout << "After Circular Left Shift: " << leftShifted << endl;
    cout << "After Circular Right Shift: " << rightShifted << endl;

    return 0;
}

```

Output:

Enter an integer: 25

Enter shift value: 2

Original Number (8-bit): 25

After Circular Left Shift: 100

After Circular Right Shift: 6