**1.print the pattern    Numbers**

**Code:**

#include <iostream>

int main() {

int rows;

std::cout << "Enter the number of rows: ";

std::cin >> rows;

for (int i = 1; i <= rows; ++i) {

for (int j = 1; j <= i; ++j) {

std::cout << j << " ";

}

std::cout << std::endl;

}

return 0;

}

**Output:**

Enter the number of rows: 4

1

1 2

1 2 3

1 2 3 4

**2.print the pattern      STAR**

**Code:**

#include <iostream>

int main() {

int rows;

std::cout << "Enter the number of rows: ";

std::cin >> rows;

for (int i = 1; i <= rows; ++i) {

for (int j = 1; j <= i; ++j) {

std::cout << "\* ";

}

std::cout << std::endl;

}

return 0;

}

**Output:**

Enter the number of rows: 4

\*

\* \*

\* \* \*

\* \* \* \*

**3.Print pascal triangle pattern nested for loop**

**Code:**

#include <iostream>

int main() {

int rows;

std::cout << "Enter the number of rows: ";

std::cin >> rows;

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

int number = 1;

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

std::cout << number << " ";

number = number \* (i - j) / (j + 1);

}

std::cout << std::endl;

}

return 0;

}

**Output:**

Enter the number of rows: 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

**4.Print diamond pattern with \* using nested for loop**

**Code:**

#include <iostream>

#include <cmath>

int main() {

int n;

std::cout << "Enter the number of rows: ";

std::cin >> n;

for (int i = 1; i <= 2 \* n - 1; i++) {

int spaces = std::abs(n - i);

int stars = 2 \* n - 1 - 2 \* spaces;

for (int j = 0; j < spaces; j++)

std::cout << " ";

for (int j = 0; j < stars; j++)

std::cout << "\*";

std::cout << std::endl;

}

return 0;

}

**Output:**

Enter the number of rows: 4

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

**5.Program to reverse the elements in an array**

**Code:**

#include <iostream>

#include <algorithm>

int main() {

int size;

std::cout << "Enter the size of the array: ";

std::cin >> size;

int arr[size];

std::cout << "Enter " << size << " elements: ";

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

std::cin >> arr[i];

}

std::reverse(arr, arr + size);

std::cout << "Reversed array: ";

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

std::cout << arr[i] << " ";

}

std::cout << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 2

Enter 2 elements: 123 678

Reversed array: 678 123

**6.Program to insert an element in an array at a specific  
position**

**Code:**

#include <iostream>

#include <vector>

int main() {

std::vector<int> arr;

int size, element, position;

std::cout << "Enter the size of the array: ";

std::cin >> size;

std::cout << "Enter " << size << " elements: ";

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

int num;

std::cin >> num;

arr.push\_back(num);

}

std::cout << "Enter the element to insert: ";

std::cin >> element;

std::cout << "Enter the position to insert (0-based indexing): ";

std::cin >> position;

if (position >= 0 && position <= size) {

arr.insert(arr.begin() + position, element);

size++;

} else {

std::cout << "Invalid position. Element cannot be inserted." << std::endl;

}

std::cout << "Array after insertion: ";

for (int num : arr) {

std::cout << num << " ";

}

std::cout << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 5

Enter 5 elements: 1 2 3 4 5

Enter the element to insert: 10

Enter the position to insert (0-based indexing): 2

Array after insertion: 1 2 10 3 4 5

**7.Program to Delete an element in an array at a specific  
position**

**Code:**

#include <iostream>

#include <vector>

int main() {

std::vector<int> arr;

int size, position;

std::cout << "Enter the size of the array: ";

std::cin >> size;

std::cout << "Enter " << size << " elements: ";

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

int num;

std::cin >> num;

arr.push\_back(num);

}

std::cout << "Enter the position of the element to delete (0-based indexing): ";

std::cin >> position;

if (position >= 0 && position < size) {

arr.erase(arr.begin() + position);

size--;

} else {

std::cout << "Invalid position. Element cannot be deleted." << std::endl;

}

std::cout << "Array after deletion: ";

for (int num : arr) {

std::cout << num << " ";

}

std::cout << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 5

Enter 5 elements: 12

23

44

55

66

Enter the position of the element to delete (0-based indexing): 2

Array after deletion: 12 23 55 66

**8.Find the sum of all elements in an array**

**Code:**

#include <iostream>

#include <vector>

#include <numeric> // for std::accumulate

int main() {

std::vector<int> arr;

int size;

std::cout << "Enter the size of the array: ";

std::cin >> size;

std::cout << "Enter " << size << " elements: ";

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

int num;

std::cin >> num;

arr.push\_back(num);

}

int sum = std::accumulate(arr.begin(), arr.end(), 0);

std::cout << "Sum of all elements in the array: " << sum << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 3

Enter 3 elements: 12 34 5

Sum of all elements in the array: 51

**9.Find the average of all elements in an array**

**Code:**

#include <iostream>

#include <vector>

#include <numeric> // for std::accumulate

int main() {

std::vector<int> arr;

int size;

std::cout << "Enter the size of the array: ";

std::cin >> size;

std::cout << "Enter " << size << " elements: ";

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

int num;

std::cin >> num;

arr.push\_back(num);

}

double average = std::accumulate(arr.begin(), arr.end(), 0.0) / size;

std::cout << "Average of all elements in the array: " << average << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 3

Enter 3 elements: 34 44

55

Average of all elements in the array: 44.3333

**10.Find the second largest element in an array**

**Code:**

#include <iostream>

#include <vector>

#include <algorithm>

int main() {

std::vector<int> arr;

int size;

std::cout << "Enter the size of the array: ";

std::cin >> size;

std::cout << "Enter " << size << " elements: ";

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

int num;

std::cin >> num;

arr.push\_back(num);

}

std::sort(arr.begin(), arr.end());

int secondLargest = arr[size - 2];

std::cout << "Second largest element in the array: " << secondLargest << std::endl;

return 0;

}

**Output:**

Enter the size of the array: 2

Enter 2 elements: 33

44

Second largest element in the array: 33