

1- Write a program that creates a queue, adds three integers (5, 6, and 7) to it, removes the first element each time, and then prints the elements of the queue.

اكتب برنامجًا يقوم بإنشاء قائمة queue وإضافة ثلاثة أعداد صحيحة (5، 6، و7) إليها، وإزالة العنصر الأول في كل مرة، ثم طباعة عناصر قائمة queue.

Output

```
5 6 7
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;

    // Add integers 5, 6, and 7 to the queue
    x.push(5);
    x.push(6);
    x.push(7);

    // Print and remove the first element from the queue
    cout << x.front() << " ";
    x.pop();

    // Print and remove the second element from the queue
    cout << x.front() << " ";
    x.pop();

    // Print and remove the third element from the queue
    cout << x.front() << " ";
    x.pop();

    return 0;
}
```

2- Write a C++ program that creates a queue, adds three integers (5, 6, and 7) to it, removes the first two elements, and then prints the first element remaining in the queue.

اكتب برنامج ++C يقوم بإنشاء قائمة queue وإضافة ثلاثة أعداد صحيحة (5، 6، و7) إليها، وإزالة العنصرين الأولين، ثم طباعة العنصر الأول المتبقي في قائمة queue.

Output

```
First element remaining in the queue: 7
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;

    // Add integers 5, 6, and 7 to the queue
    x.push(5);
    x.push(6);
    x.push(7);

    // Remove the first two elements from the queue
    x.pop();
    x.pop();

    // Print the first element remaining in the queue
    cout << "First element remaining in the queue: " << x.front() << endl;

    return 0;
}
```

3- Write a C++ program that creates a queue, adds three integers (5, 6, and 7) to it, and then prints and removes each element from the queue using a while loop.

اكتب برنامج ++C يقوم بإنشاء قائمة queue وإضافة ثلاثة أعداد صحيحة (5، 6، و 7) إليها، ثم طباعة كل عنصر من قائمة queue وإزالته باستخدام حلقة while.

Output

5 6 7

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;

    // Add integers 5, 6, and 7 to the queue
    x.push(5);
    x.push(6);
    x.push(7);

    // Print and remove each element from the queue using a while loop
    while (!x.empty()) {
        cout << x.front() << " ";
        x.pop();
    }

    return 0;
}
```

4- Write a C++ program that creates a queue, adds three integers (5, 6, and 7) to it, and then prints and removes each element from the queue using a while loop. After processing all elements, print the total number of elements processed.

اكتب برنامج ++C يقوم بإنشاء قائمة queue وإضافة ثلاثة أعداد صحيحة (5، 6، و 7) إليها، ثم طباعة كل عنصر من قائمة queue وإزالته باستخدام حلقة while. بعد معالجة جميع العناصر، قم بطباعة العدد الإجمالي للعناصر التي تمت معالجتها.

Output

```
5 6 7
Total elements processed: 3
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;
    int count = 0;

    // Add integers 5, 6, and 7 to the queue
    x.push(5);
    x.push(6);
    x.push(7);

    // Process and remove each element from the queue using a while loop
    while (!x.empty()) {
        count++;
        cout << x.front() << " ";
        x.pop();
    }

    // Print the total number of elements processed
    cout << endl << "Total elements processed: " << count << endl;

    return 0;
}
```

5- Write a program in C++ that demonstrates the usage of a queue. Follow the instructions below:

Create a queue named x.

Push three integer values (5, 6, and 7) into the queue x.

Process all elements in the queue by printing them.

Count the total number of elements processed.

Output the total number of elements processed.

Output the size of the queue after processing.

اكتب برنامجًا بلغة ++C يوضح استخدام قائمة queue. اتبع التعليمات التالية:

قم بإنشاء قائمة انتظار باسم x.

ادفع ثلاث قيم صحيحة (5، 6، و7) إلى قائمة queue x.

معالجة جميع العناصر الموجودة في قائمة queue عن طريق طباعتها.

حساب العدد الإجمالي للعناصر التي تمت معالجتها.

إخراج العدد الإجمالي للعناصر التي تمت معالجتها.

إخراج حجم قائمة queue بعد المعالجة.

Output

```
5 6 7
Total elements processed: 3
Size of the queue after processing: 0
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;
    int count = 0;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Process all elements in the queue and print them
    while (!x.empty()) {
        count++; // Increment the count for each element processed
        cout << x.front() << " "; // Print the front element of the queue
        x.pop(); // Remove the front element from the queue
    }

    // Output the total number of elements processed
    cout << endl << "Total elements processed: " << count << endl;

    // Output the size of the queue after processing
    cout << "Size of the queue after processing: " << x.size() << endl;

    return 0;
}
```

6- Write a C++ program that demonstrates the usage of a queue. Follow the instructions below:

Create a queue named x.

Push three integer values (5, 6, and 7) into the queue x.

Print each element in the queue by incrementing its value by 1.

اكتب برنامج ++C يوضح استخدام قائمة الانتظار. اتبع التعليمات التالية:

قم بإنشاء قائمة queue باسم x.

ادفع ثلاث قيم صحيحة (5، 6، و 7) إلى قائمة x.queue.

اطبع كل عنصر في قائمة queue بزيادة قيمته بمقدار 1.

Output

6 7 8

Solution

```
// www.gammal.tech

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

int main() {
    // Create a queue named x
    queue<int> x;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Process all elements in the queue and print them after incrementing by 1
    while (!x.empty()) {
        // Print the front element of the queue after incrementing by 1
        cout << x.front() + 1 << " ";
        x.pop(); // Remove the front element from the queue
    }

    return 0;
}
```

7- Write a C++ program that demonstrates the usage of two queues. Follow the instructions below:

Create two queues named x and y.

Push three integer values (5, 6, and 7) into the queue x.

Transfer all elements from queue x to queue y.

Print all elements in queue y.

اكتب برنامج ++C يوضح استخدام قائمتين من queue . اتبع التعليمات التالية:

قم بإنشاء قائمتين بالاسم x و y.

ادفع ثلاث قيم صحيحة (5، 6، و 7) إلى قائمة x queue.

نقل كافة العناصر من قائمة الانتظار x إلى قائمة y queue.

طباعة جميع العناصر في قائمة الانتظار y.

Output

5 6 7

Solution

```
// www.gammal.tech

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

int main() {
    // Create two queues named x and y
    queue<int> x;
    queue<int> y;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Transfer all elements from queue x to queue y
    while (!x.empty()) {
        y.push(x.front()); // Push the front element of queue x into queue y
        x.pop();           // Remove the front element from queue x
    }

    // Print all elements in queue y
    while (!y.empty()) {
        cout << y.front() << " "; // Print the front element of queue y
        y.pop();                  // Remove the front element from queue y
    }

    return 0;
}
```

8- Write a C++ program that demonstrates the usage of two queues. Follow the instructions below:

Create two queues named x and y.

Push three integer values (5, 6, and 7) into the queue x.

Transfer all elements from queue x to queue y.

Print the front element of queue x after the transfer operation

اكتب برنامج ++C يوضح استخدام قائمتين من queue . اتبع التعليمات التالية:

قم بإنشاء قائمتين بالاسم x و y.

ادفع ثلاث قيم صحيحة (5، 6، و 7) إلى قائمة x.queue.

نقل كافة العناصر من قائمة الانتظار x إلى قائمة y.queue.

اطبع العنصر الأمامي لقائمة الانتظار x بعد عملية النقل

Output

0

Solution

```
// www.gammal.tech

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

int main() {
    // Create two queues named x and y
    queue<int> x;
    queue<int> y;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Transfer all elements from queue x to queue y
    while (!x.empty()) {
        y.push(x.front()); // Push the front element of queue x into queue y
        x.pop();           // Remove the front element from queue x
    }

    cout << x.front() << endl;

    return 0;
}
```

9- Write a C++ program that demonstrates the usage of two queues. Follow the instructions below:

Create two queues named x and y.

Push three integer values (5, 6, and 7) into the queue x.

Transfer all elements from queue x to queue y.

Push three additional integer values (8, 9, and 10) into queue y.

Print all elements of queue y after the transfer and additional push operations.

اكتب برنامج ++C يوضح استخدام قائمتين من queue . اتبع التعليمات التالية:

قم بإنشاء قائمتين بالاسم x و y.

ادفع ثلاث قيم صحيحة (5، 6، و 7) إلى queue x.

نقل كافة العناصر من قائمة الانتظار x إلى قائمة queue y.

push ثلاث قيم صحيحة إضافية (8 و 9 و 10) إلى قائمة queue y.

اطبع جميع عناصر قائمة queue y بعد النقل وعمليات push الإضافية.

Output

```
Elements of queue y: 5 6 7 8 9 10
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create two queues named x and y
    queue<int> x;
    queue<int> y;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Transfer all elements from queue x to queue y
    while (!x.empty()) {
        y.push(x.front()); // Push the front element of queue x into queue y
        x.pop();           // Remove the front element from queue x
    }

    // Push additional integer values 8, 9, and 10 into queue y
    y.push(8);
    y.push(9);
    y.push(10);

    // Print all elements of queue y
    cout << "Elements of queue y: ";
    while (!y.empty()) {
        cout << y.front() << " "; // Print the front element of queue y
        y.pop();                   // Remove the front element from queue y
    }

    return 0;
}
```

10- Write a C++ program that demonstrates the usage of two queues. Follow the instructions below:

Create two queues named x and y.

Push three integer values (5, 6, and 7) into the queue x.

Transfer all elements from queue x to queue y.

Push three additional integer values (8, 9, and 10) into queue y.

Output the sizes of queues x and y after the transfer and additional push operations.

اكتب برنامج ++C يوضح استخدام قائمتين من queue . اتبع التعليمات التالية:

قم بإنشاء قائمتين بالاسم x و y.

ادفع ثلاث قيم صحيحة (5، 6، و 7) إلى قائمة queue x.

نقل كافة العناصر من قائمة الانتظار x إلى قائمة queue y.

ادفع ثلاث قيم صحيحة إضافية (8 و 9 و 10) إلى قائمة الانتظار y.

إخراج أحجام قوائم queue x و y بعد النقل وعمليات push الإضافية.

Output

```
Size of queue x: 0
Size of queue y: 6
```

Solution

```
// www.gammal.tech

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

int main() {
    // Create two queues named x and y
    queue<int> x;
    queue<int> y;

    // Push integer values 5, 6, and 7 into the queue x
    x.push(5);
    x.push(6);
    x.push(7);

    // Transfer all elements from queue x to queue y
    while (!x.empty()) {
        y.push(x.front()); // Push the front element of queue x into queue y
        x.pop();           // Remove the front element from queue x
    }

    // Push additional integer values 8, 9, and 10 into queue y
    y.push(8);
    y.push(9);
    y.push(10);

    // Output the sizes of queues x and y
    cout << "Size of queue x: " << x.size() << endl;
    cout << "Size of queue y: " << y.size() << endl;

    return 0;
}
```