

بسم الله الرحمن الرحيم

حل المشكلة ١

اكتب برنامج يطبع جدول الضرب من ١ الى ١٠ كما يلي :

Problem:

Write a program print the multiplication table from 1 to 10 as follows:

Multipliacion Table From 1 to 10										
	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

ماذا يعمل ؟	Function & Procedures ...	م
طباعة عنوان الجدول	<code>void PrintTableHeader</code>	١
طباعة الفاصل " "	<code>string ColumnSperator</code>	٢
طباعة جدول الضرب مع التنسيق	<code>void PrintMultiplicationTable</code>	٣

```

#include<iostream>
using namespace std;

void PrintTableHeader()
{
    // طباعة عنوان الجدول والأرقام العلوية والخط
    cout << "\n\n\t\t\t Multipliacion Table From 1 to 10\n\n";
    cout << "\t";
    for (int i = 1; i <= 10; i++)
    {
        cout << i << "\t"; // Print ( 1    2    3 ...)
    }
    cout <<
    "\n-----\n";
}

string ColumnSperator(int i)
{
    // طباعة الفاصل "|" الذي بين الأرقام العمودية وجدول الضرب
    if (i < 10) // الفاصل "|" مع الرقم بعد الرقم مع
        return " |"; // اطبع ٣ مسافات
    else
        return " |"; // اطبع ٩ مسافتين
}

void PrintMultiplicationTable()
{
    // استدعاء عنوان الجدول
    PrintTableHeader();

    for (int i = 1; i <= 10; i++) // للأسطر
    {
        // طباعة حلقة التكرار للأرقام العمودية من ١ الى ١٠ مع استدعاء الفاصل "|" مع كل تكرار
        cout << " " << i << ColumnSperator(i) << "\t"; // Print ( 1 | )
        // ثم الدخول الى حلقة تكرار جديدة لطباعة جدول الضرب
        for (int j = 1; j <= 10; j++) // للصف
        {
            cout << j * i << "\t"; // Print ( 1 | 1 2 .... )
        }
        cout << endl;
    }
}

int main()
{
    PrintMultiplicationTable();
}

```

حل المشكلة ٢

اكتب برنامج لقراءة رقم ، واطبع الأرقام الأولية من ١ الى الرقم

المدخلات : ١٠ المخرجات : ١ ٢ ٣ ٥ ٧

ماذا يعمل ؟	Function & Procedures ...	م
حفظ القيم المطلوبة	<code>enum enPrimNotPrime</code>	١
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	٢
معادلة لاختيار الرقم المدخل إما عدد أولي أو لا	<code>enPrimNotPrime CheckPrime</code>	٣
طباعة النتيجة من معادلة الاختيار	<code>void PrintPrimeNumbersFrom1ToNumber</code>	٤

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

enum enPrimNotPrime { Prime = 1, NotPrime = 2 };

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 10
    } while (Number <= 0);

    return Number;
}

enPrimNotPrime CheckPrime(int Number)
{
    // Number = 10
    int M = round(Number / 2); // 10 / 2 = 5

    for (int Counter = 2; Counter <= M; Counter++) // 2 <= 5 ?
    {
        if (Number % Counter == 0) // 10 % 2 == 0 ? Yes , is (Not Prime)
            return enPrimNotPrime::NotPrime;
    }
    return enPrimNotPrime::Prime; // 11 % 2 == 0 ? No , is (Prime)
}

void PrintPrimeNumbersFrom1ToNumber(int Number)
{
    cout << "\n Prime Numbers from " << 1 << " To " << Number << " are :
\n";

    for (int i = 1; i <= Number; i++) // 1 <= 10 ? التكرار في حلقة التكرار
    {
        if (CheckPrime(i) == enPrimNotPrime::Prime)
            // CheckPrime(1 ...2 3 to 10) == enPrimNotPrime::Prime
            cout << i << endl; // Print ( 1 \n 2 \n 3 \n 5 ....)
    }
}

int main()
{
    PrintPrimeNumbersFrom1ToNumber(ReadPositiveNumber("Pleas enter a
positive number "));
}
```

حل المشكلة ٣

اكتب برنامج للتحقق مما إذا كان الرقم صحيحاً أم لا ؟

ملاحظة : الرقم المثالي = مجموع (جميع القواسم للرقم)

مثال : $28 = 1 + 2 + 4 + 7 + 14$ $6 = 1 + 2 + 3$

إخال : ٢٨ ١٢ المخرجات : 28 is Perfect 12 is not Perfect

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int ReadPositiveNumber</code>	قراءة رقم من المستخدم
٢	<code>bool IsPerfectNumber</code>	معادلة لاختيار، هل الرقم المدخل رقم مثالي أو لا ؟ true or false
٣	<code>void PrintResult</code>	طباعة النتيجة من معادلة الاختيار

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 28
    } while (Number <= 0);

    return Number;
}

bool IsPerfectNumber(int Number)
{
    // Number = 28
    int Sum = 0; // الأفضل يكون لكل متغير قيمة
    for (int i = 1; i < Number; i++) // 1 < 28 ?
    {
        if (Number % i == 0) // 28 % 1 = 0 ?; 28 % 2 = 0 ?...
            Sum += i; // 1 + 2 + 4 ...
    }
    return Number == Sum; // 28 == 28 ? , true (or false )
}

void PrintResult(int Number)
{
    if (IsPerfectNumber(Number)) // true(28)
        cout << "\n " << Number << " Is Perfect Number \n";
    else // false(12)
        cout << "\n " << Number << " Is Not Perfect Number \n";
}

int main()
{
    PrintResult(ReadPositiveNumber("Pleas enter a positive number "));
}
```

حل المشكلة ٤

اكتب برنامج لطباعة جميع الأرقام المثالية من ١ الى الرقم المدخل

المدخلات : ٥٠٠ المخرجات : ٦ ٢٨ ٤٩٦

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int ReadPositiveNumber</code>	قراءة رقم من المستخدم
٢	<code>bool IsPerfectNumber</code>	معادلة لاختيار، هل الرقم المدخل رقم مثالي أو لا ؟ true or false
٣	<code>void PrintPerfectNumberFrom1ToNum</code>	طباعة النتيجة من معادلة الاختيار (الصحيح فقط)

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 500

    } while (Number <= 0);

    return Number;
}

bool IsPerfectNumber(int Number)
{
    // Number = 500
    int Sum = 0; // الأفضل يكون لكل متغير قيمة
    for (int i = 1; i < Number; i++) // 1 < 500 ?
    {
        if (Number % i == 0) // 500 % 1 = 0 ?; 500 % 2 = 0...(to 500
        < 499 ). 499 % 1 = 0 ...
            Sum += i; // 1 + 2 + 4 ...
    }
    return Number == Sum; // 500 == 592 is false .. ( true = 6 , 28 ,
    496 )
}

void PrintPerfectNumberFrom1ToNum(int Number)
{
    cout << "\n";
    for (int i = 1; i < Number; i++) // 1 < 500 ?
    {
        if (IsPerfectNumber(i)) // IsPerfectNumber(500) ? المقبول إذا كان
        الجواب صحيح فقط
            cout << i << endl; // Print ( true = 6 , 28 , 496 )
    }
}

int main()
{
    PrintPerfectNumberFrom1ToNum(ReadPositiveNumber("Pleas enter a
    positive number ? "));
}
```

حل المشكلة ٥

اكتب برنامج لقراءة رقم وطباعته بترتيب معكوس

المدخلات : ١٢٣٤

المخرجات :

٤

٣

٢

١

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int</code> ReadPositiveNumber	١
طباعة الرقم بترتيب معكوس	<code>void</code> PrintDigits	٢

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 1234

    } while (Number <= 0);

    return Number;
}

void PrintDigits(int Number)
{
    // Number = 1234
    int Remainder = 0;
    while (Number > 0) // 1234 > 0 ? | 123 > 0 ? ....
    {
        Remainder = Number % 10; // 1234 % 10 = 4 | 123 % 10 = 3 ....
        Number = Number / 10; // (int) 1234 / 10 = 123 | 123 / 10 = 12
        ....
        cout << Remainder << endl; // Print ( 4 \n 3 \n 2 \n 1 )
    }
}

int main()
{
    PrintDigits(ReadPositiveNumber("Please enter a positive Number ?"));
}
```

حل المشكلة ٦

اكتب برنامج لقراءة رقم وطباعة مجموع أرقامه

المخرجات : 10 Sum Of Digits

المدخلات : ١٢٣٤

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
طباعة مجموع الأرقام	<code>int PrintSumOfDigits</code>	٢

```

#include <iostream>
using namespace std;

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 1234
    } while (Number <= 0);

    return Number;
}

int PrintSumOfDigits(int Number)
{
    // Number = 1234
    int Sum = 0, Remainder = 0;
    while (Number > 0) // 1234 > 0 ? | 123 = 0 ? ...
    {
        Remainder = Number % 10; // 1234 % 10 = 4 | 123 % 10 = 3 ....
        Number = Number / 10; //(int) 1234 / 10 = 123 | 123 / 10 = 12
        ....
        Sum = Sum + Remainder; // 0 + 4 = 4 | 4 + 3 = 7 ....
    }
    return Sum; // Sum = 10
}

int main()
{
    cout << "\n Sum Of Digits = "
        << PrintSumOfDigits(ReadPositiveNumber("Pleas enter a positie
Number ?"))
        << endl;
}

```

حل المشكلة ٧

اكتب برنامج لقراءة رقم وطباعته معكوسا

المخرجات : ٤٣٢١

المدخلات : ١٢٣٤

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
طباعة الرقم بشكل معكوس	<code>int ReverseNumbers</code>	٢

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 1234
    } while (Number <= 0);

    return Number;
}

int ReverseNumbers(int Number)
{
    // Number = 1234
    int Remainder = 0 , Number2 = 0;
    while (Number > 0) // 1234 > 0 ?
    {
        Remainder = Number % 10; // 1234 % 10 = 4 | 123 % 10 = 3 ....
        Number = Number / 10; //(int) 1234 / 10 = 123 | 123 / 10 = 12
        ....
        Number2 = Number2 * 10 + Remainder;
    }
    // 0 * 10 + 4 = 4 | 4 * 10 + 3 = 43 .....
    return Number2; // Number2 = 4321
}

int main()
{
    cout << "\n Reverse is : \n "
        << ReverseNumbers(ReadPositiveNumber("Pleas enter a positie
Number ?"));
}
```


حل المشكلة ٨

اكتب برنامج يقرأ رقم وقراءة خانة من الرقم المدخل ، ثم اطبع تكرار خانة

المدخلات : ١٢٢٣٢٢٢ \ ٢ المخرجات : Digit 2 Frequency is 5 Time(s).

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int ReadPositiveNumber</code>	قراءة رقمين من المستخدم (استدعاؤه مرتين)
٢	<code>int CountDigitFrequency</code>	طباعة عدد تكرار الرقم الأول

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number <= 0);

    return Number;
}

int CountDigitFrequency(int Number, short DigitToCheck)
{
    // Number = 1223222 , DigitToCheck = 2
    int Remainder = 0;
    short FreqCount = 0;
    while (Number > 0) // 1223222 > 0 ? | 122322 > 0 ? .....
    {
        Remainder = Number % 10;
        // 1223222 % 10 = 2 | 122322 % 10 = 2 ....

        Number = Number / 10;
        // (int) 1223222 / 10 = 122322 | 122322 / 10 = 12232 ....

        if (DigitToCheck == Remainder) // 2 == 2 ? | 2 == 2 ?
        {
            FreqCount++; // 0 + 1 = 1 | 1 + 1 = 2 ...
        }
    }
    return FreqCount; // FreqCount = 5
}

int main()
{
    int Number = ReadPositiveNumber("Pleas enter the main Number ?");
    // 1223222
    short DigitToCheck = ReadPositiveNumber("Pleas enter one digit to
check ?"); // 2
    // استخدام ReadPositiveNumber أكثر من مرة ، لقراءة متغيرين مختلفين

    cout << "\n Digit " << DigitToCheck << " Frequency is "
        << CountDigitFrequency(Number, DigitToCheck) << " Time(s).
\n";
    // Print ( Digit 2 Frequency is 5 Time(s).
    return 0;
}
```

حل المشكلة ٩

اكتب برنامج يقرأ رقم ، ثم يطبع تكرار كل رقم

المدخلات : ١٢٢٣٢٢٢

المخرجات : Digit 1 Frequency is 1 Time(s).

Digit 2 Frequency is 5 Time(s).

Digit 3 Frequency is 1 Time(s).

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int</code> ReadPositiveNumber	١
عدد تكرار الأرقام	<code>int</code> CountDigitFrequency	٢
طباعة تكرار جميع الأرقام من ٠ الى ٩	<code>void</code> PrintAllDigitsFrequency	٣

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 1223222
    } while (Number <= 0);

    return Number;
}

int CountDigitFrequency(int Number, short DigitToCheck)
{
    // Number = 1223222
    // DigitToCheck = 0 | DigitToCheck = 1 ....
    int Remainder = 0;
    short FreqCount = 0;
    while (Number > 0) // 1223222 > 0 ? | 122322 > 0 ? .....
    {
        Remainder = Number % 10;
        // 1223222 % 10 = 2 | 122322 % 10 = 2 ....
        Number = Number / 10;
        // (int) 1223222 / 10 = 122322 | 122322 / 10 = 12232 ....
        if (DigitToCheck == Remainder) // 0 == 2 ? | 1 == 2 ? | 2 ==
        2 ....
        {
            FreqCount++; // 0 + 1 = 1 || 0 + 5 = 5 ...
        }
    }
    return FreqCount;
    // FreqCount = 0 DigitToCheck = 0 | FreqCount = 1 DigitToCheck = 1 |
    FreqCount = 5 DigitToCheck = 2 ....
}
```

```

void PrintAllDigitsFrequency(int Number)
{
    // Number = 1223222
    for (int i = 0; i < 10; i++) // 0 < 10 ? | 1 < 10 ? ....
    {
        short DigitFrequency = 0;
        DigitFrequency = CountDigitFrequency(Number, i);
        // إعادة استخدام الكود السابق | CountDigitFrequency ( 1223222 , 0 ) |
        CountDigitFrequency ( 1223222 , 1 )

        if (DigitFrequency > 0) // 0 > 0 ? | 1 > 0 ? ....
            // إذا كان الرقم موجود DigitFrequency اطبع الرقم وكم مرة تكرر
            {
                cout << "\n Digit " << i << " Frequency is "
                     << DigitFrequency << " Time(s). \n";
            }
    }
}

int main()
{
    int Number = ReadPositiveNumber("Pleas enter the main Number ?");
    PrintAllDigitsFrequency(Number);
    return 0;
}

```

حل المشكلة ١٠

اكتب برنامج لقراءة رقم وطباعته بالترتيب من اليسار إلى اليمين

المدخلات : ١٢٣٤

المخرجات : ١ ٢ ٣ ٤

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
قلب الرقم بشكل معكوس	<code>int ReverseNumbers</code>	٢
طباعة الرقم المعكوس بشكل صحيح (كالرقم المدخل)	<code>void PrintDigits</code>	٣

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 1234
    } while (Number <= 0);

    return Number;
}

int ReverseNumbers(int Number)
{
    // Number = 1234
    int Remainder = 0;
    int Number2 = 0;
    while (Number > 0) // 1234 > 0 ? | 123 > 0 ? ....
    {
        Remainder = Number % 10; // 1234 % 10 = 4 | 123 % 10 = 3
        ....
        Number /= 10; // 1234 / 10 = 123 | 123 / 10 = 12 ....
        Number2 = Number2 * 10 + Remainder;
        // 0 * 10 + 4 = 4 | 4 * 10 + 3 = 43 ....
    }
    return Number2; // Number2 = 4321
}

void PrintDigits(int Number)
{
    // Number = 4321
    int Remainder = 0;
    while (Number > 0) // 4321 > 0 ? | 432 > 0 ? .....
    {
        Remainder = Number % 10; // 1234 % 10 = 4 | 123 % 10 = 3
        ....
        Number /= 10; // 4321 / 10 = 432 | 432 / 10 = 43 ....
        cout << Remainder << endl; // 1 \n 2 \n 3 \n 4
    }
}

int main()
{
    PrintDigits(ReverseNumbers(ReadPositiveNumber("Please enter a positive Number ?")));
}
```

حل المشكلة ١١

اكتب برنامج لقراءة رقم ومعرفة ما إذا كان الرقم يقرأ من اليمين الى اليسار والعكس ؟

المدخلات : ١٢٣٤ \ ١٢٣٢١

المخرجات : No , it is NOT a Palindrome number

Yes , it is a Palindrome number

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
قلب الرقم بشكل معكوس	<code>int ReverseNumbers</code>	٢
هل الرقم المدخل == عكس الرقم ؟ (true or false)	<code>bool IsPalindrome</code>	٣

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 12321
    } while (Number <= 0);

    return Number;
}

int ReverseNumbers(int Number)
{
    // Number = 1234
    int Remainder = 0;
    int Number2 = 0;
    while (Number > 0) // 12321 > 0 ? | 1232 > 0 ? ....
    {
        Remainder = Number % 10; // 12321 % 10 = 1 | 1232 % 10 = 2
        ....
        Number /= 10; // 12321 / 10 = 1232 | 1232 / 10 = 123 ....
        Number2 = Number2 * 10 + Remainder;
        // 0 * 10 + 1 = 1 | 1 * 10 + 2 = 12 ....
    }
    return Number2; // Number2 = 12321
}

bool IsPalindrome(int Number)
{
    // Number = 12321
    // Number = 1234
    return Number == ReverseNumbers(Number);
    // 12321 == 12321 ? true , ( 1234 == 4321 ? false )
}

int main()
{
    if (IsPalindrome(ReverseNumbers(ReadPositiveNumber("Pleas enter a
positie Number ?"))))
        cout << "\n Yes , it is a Palindrome number \n";
    // true 12321
    else
        cout << "\n No , it is NOT a Palindrome number \n";
    // false 1234
}
```

حل المشكلة ١٢

اكتب برنامج لقراءة رقم وطباعة نمط مقلوب . على النحو التالي

المدخلات : ٣ \ ٥

المخرجات : ٣٣٣ ٢٢ ١

٥٥٥٥٥ ٤٤٤٤ ٣٣٣ ٢٢ ١

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int ReadPositiveNumber</code>	قراءة رقم من المستخدم
٢	<code>void PrintInvertedNumberPattern</code>	تكرار طباعة الرقم المدخل بعدده الى رقم ١

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 3
    } while (Number <= 0);

    return Number;
}

void PrintInvertedNumberPattern(int Number)
{
    // Number = 3
    cout << "\n";
    // حلقة التكرار تكون للأسطر
    for (int i = Number; i >= 1; i--) // 3 >= 1 ? | 2 >= 1 ? ...
    {
        for (int j = 1; j <= i; j++)
        // 1 <= 3 ? | 2 <= 3 ? ... | 1 <= 2 ? ....
        {
            // حلقة التكرار هذه للصفوف
            cout << i; // Print ( 333 \n 22 \n 1 )
        }
        cout << endl;
    }
}

int main()
{
    PrintInvertedNumberPattern(ReadPositiveNumber("Pleas enter a positie
Number ?"));
}
```

حل المشكلة ١٣

اكتب برنامج لقراءة رقم وطباعة نمط بشكل صحيح. على النحو التالي

المدخلات: ٣ \ ٥

المخرجات: ١ ٢٢ ٣٣

١ ٢٢ ٣٣٣ ٤٤٤٤ ٥٥٥٥٥

م	Function & Procedures ...	ماذا يعمل؟
١	<code>int ReadPositiveNumber</code>	قراءة رقم من المستخدم
٢	<code>void PrintInvertedNumberPattern</code>	تكرار طباعة الرقم من ١ الى الرقم المدخل بعده

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

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number; // 3
    } while (Number <= 0);

    return Number;
}

void PrintNumberPattern(int Number)
{
    // Number = 3
    cout << "\n";
    // حلقة التكرار تكون للأسطر
    for (int i = 1; i <= Number; i++) // 1 <= 3 ? | 2 <= 3 ? ...
    {
        for (int j = 1; j <= i; j++) // 1 <= 1 ? | 1 <= 2 ? | 2 <= 2
        ? ....
        {
            // حلقة التكرار هذه للصفوف
            cout << i; // Print ( 1 \n 22 \n 333 )
        }
        cout << endl;
    }
}

int main()
{
    PrintNumberPattern(ReadPositiveNumber("Pleas enter a positie Number
?"));
}
```

حل المشكلة ١٤

اكتب برنامج لقراءة رقم وطباعة نمط الحرف المقلوب . على النحو التالي

المدخلات : ٣ \ ٥

المخرجات : CCC BB A

EEEE DDDD CCC BB A

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
تكرار طباعة الحرف المدخل بعدد ترتيبه في الحروف من رقم الحرف المدخل الى A	<code>void PrintInvertedLetterPattern</code>	٢

```
#include <iostream>

using namespace std;

int ReadPositiveNumber(string Message)
{
    int Number;

    do
    {
        cout << Message << endl;
        cin >> Number; // 3

    } while (Number < 0);
    return Number;
}

void PrintInvertedLetterPattern(int Number)
{
    // Number = 3 ( عدد الأحرف أو الأسطر )
    cout << "\n";

    for (int i = 65 + Number - 1 ; i >= 65; i--)
    // 67 >= 65 ? || 66 >= 65 ? ...
    {
        for (int j = 1 ; j <= Number - ( 65 + Number - 1 - i ); j++)
        // 1 <= 3 [= 3 - "0" ( 65 + 3 - 1 - 67 ) ] || 2 <= 3 [= 3 - "0" ( 65 + 3
        // - 1 - 67 ) ] ...
        {
            cout << char(i); // CCC \n BB \n A
        }
        cout << endl;
    }
}

int main()
{
    PrintInvertedLetterPattern(ReadPositiveNumber("Pleas enter a
    Positive Number ?"));
}
```


حل المشكلة ١٥

اكتب برنامج لقراءة رقم وطباعة نمط الحرف بشكل صحيح . على النحو التالي

المدخلات : ٣ \ ٥

المخرجات : A BB CCC

A BB CCC DDDD EEEEE

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم من المستخدم	<code>int ReadPositiveNumber</code>	١
تكرار طباعة الحرف المدخل بعدد ترتيبه في الحروف من الحرف A الى رقم الحرف المدخل	<code>void PrintLetterPattern</code>	٢

```
#include <iostream>

using namespace std;

int ReadPositiveNumber(string Message)
{
    int Number;

    do
    {
        cout << Message << endl;
        cin >> Number; // 3

    } while (Number < 0);
    return Number;
}

void PrintLetterPattern(int Number)
{
    // Number = 3 ( عدد الأحرف أو الأسطر )
    cout << "\n";

    for (int i = 65 ; i <= 65 + Number - 1; i++)
    // 65 <= "67" (65 + 3 - 1) ? || 66 <= "67" (65 + 3 - 1) ? ...
    {
        for (int j = 1 ; j <= i - 65 + 1; j++) // 1 <= "1" (65 - 65
        + 1) ? || 1 <= "2" (66 - 65 + 1) ? || 2 <= "2" (66 - 65 + 1) ? ...
        {
            cout << char(i); // A \n BB \n CCC
        }
        cout << endl;
    }
}

int main()
{
    PrintLetterPattern(ReadPositiveNumber("Pleas enter a Positive Number
?"));
}
```

حل المشكلة ١٦

اكتب برنامج لقراءة رقم وطباعة نمط الحرف بشكل صحيح . على النحو التالي

المخرجات : AAA | AAB | AAC | | ZZZ

ماذا يعمل ؟	Function & Procedures ...	م
طباعة الحروف من AAA الى ZZZ	<code>void PrintWordsFromAAAtZZZ</code>	١

```
#include <iostream>

using namespace std;

void PrintWordsFromAAAtZZZ()
{
    cout << "\n";
    string Word = "";

    for (int i = 65; i <= 90 ; i++) // 65 <= 90 ? | 66 <= 90 ? .....
    {
        for (int j = 65; j <= 90 ; j++) // 65 <= 90 ? | 66 <= 90 ?
        .....
        {
            for (int k = 65; k <= 90; k++) // 65 <= 90 ? | 66 <= 90
            ? .....
            {
                Word = Word + char(i); // A ...
                Word = Word + char(j); // A | B ...
                Word = Word + char(k); // A , B ... | A , B

                cout << Word << endl; // AAA | AAB .... | ZZZ

                Word = ""; // إرجاع المتغير الى "فارغ" لاشيء
            }
        }
        cout << "\n-----\n";
    }
}

int main()
{
    PrintWordsFromAAAtZZZ();
}
```

حل المشكلة ١٧

اكتب برنامج لتخمين كلمة المرور ، مكونة من ٣ أحرف (كلها كبيرة)

المدخلات : AAF

المخرجات :

Trial [1] : AAA

Trial [2] : AAB

Trial [3] : AAC

Trial [4] : AAD

Trial [5] : AAE

Trial [6] : AAF

Password is : AAF

Found after 6 Trial(s)

ماذا يعمل ؟	Function & Procedures ...	م
قراءة الرقم السري (حروف)	<code>string</code> ReadPassword	١
البحث عن الرقم السري	<code>bool</code> GuessPassword	٢

```

#include <iostream>

using namespace std;

string ReadPassword( )
{
    string Password;
    cout << "Pleas enter a 3-Letter Password (all capital) ? \n";
    cin >> Password; // AAF
    return Password;
}

bool GuessPassword(string OriginalPassword)
{
    // OriginalPassword = AAF
    cout << "\n";
    string Word = "";
    int Counter = 0;

    for (int i = 65; i <= 90 ; i++) // 65 <= 90 ? | 66 <= 90 ? .....
    {
        for (int j = 65; j <= 90 ; j++) // 65 <= 90 ? | 66 <= 90 ?
        .....
        {
            for (int k = 65; k <= 90; k++) // 65 <= 90 ? | 66 <= 90
            ? .....
            {
                Counter++; // عدد المحاولات

                Word = Word + char(i); // A ...
                Word = Word + char(j); // A | B ...
                Word = Word + char(k); // A , B ... | A , B

                cout << "Trial [" << Counter << "] : " << Word
                << endl; // print ( Trial [1] : AAA | Trial [2] : AAB ...
                if (Word == OriginalPassword)
                // AAA == AAF ? | AAB == AAF ? ...
                {
                    cout << "\nPassword is : " <<
                    OriginalPassword << endl; // print ( Password is : AAF )
                    cout << "Found after " << Counter << "
                    Trial(s)\n"; // print ( Found after 6 Trial(s)
                    return true; // اخرج من حلقة التكرار + إرجاع قيمة
                }

                Word = ""; // إرجاع المتغير الى "فارغ" لاشيء
            }
        }
    }

    return false;
}

int main()
{
    GuessPassword(ReadPassword());
}

```

حل المشكلة ١٨

اكتب برنامج لقراءة النص وتشفيره وفك تشفيره

المدخلات : Saeed

المخرجات :

Text Before Encryption : Saeed

Text After Encryption : Ucggf

Text After Decryption : Saeed

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>string</code> ReadText	قراءة نص من المستخدم
٢	<code>string</code> EncryptText	النص & مفتاح التشفير = نص مشفر
٣	<code>string</code> DecryptionText	النص المشفر & مفتاح التشفير = فك التشفير

```

#include <iostream>
#include <string>

using namespace std;

string ReadText()
{
    string Text;
    cout << "Pleas enter Text ? \n";
    getline(cin, Text); // Saeed
    return Text;
}

string EncryptText(string Text, short EncryptionKey)
{
    // Text = Saeed
    // EncryptionKey = 2

    for (int i = 0; i <= Text.length(); i++)
    { // Text.length() ( إنديكس ) موقع الحرف : مجزئة النص حرف حرف
        Text[i] = char((int) Text[i] + EncryptionKey);
        // Text[i] == (int) Text[S] == Text[83] + 2 == (char) Text[85] << U[0];
        .....
    }
    return Text; // Ucggf
}

string DecryptionText(string Text, short EncryptionKey)
{
    for (int i = 0; i <= Text.length(); i++)
    {
        Text[i] = char((int)Text[i] - EncryptionKey);
        // Text[i] == (int) Text[U] == Text[85] - 2 == (char) Text[83] << S[0];
        .....
    }
    return Text; // Saeed
}

int main()
{
    const short EncryptionKey = 2;

    string Text = ReadText();
    string TextAfterEncryption = EncryptText(Text, EncryptionKey);
    string TextAfterDecryption = DecryptionText(TextAfterEncryption,
EncryptionKey);

    cout << endl << "Text Before Encryption : " << Text << endl;
    cout << "Text After Encryption : " << TextAfterEncryption << endl;
    cout << "Text After Decryption : " << TextAfterDecryption << endl;
}

```

syntax	
#include <cstdlib>	المكتبة التي تضم الرقم العشوائي
srand((unsigned)time(NULL));	لابد من استدعاء هذا الكود في (main) ليعمل rand() فيعطي رقم عشوائي مختلف مع كل مرة
rand()	يعطي رقم عشوائي : عادة يكون الرقم كبير ، حجمه مثل int

اكتب برنامجا لطباعة ٣ أرقام عشوائية من ١ الى ١٠

المخرجات :

٤

٩

١

ماذا يعمل ؟	Function & Procedures ...	م
رقم عشوائي	int RandomNumber	١

```
#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    // rand() int حجمه مثل ٨٧٥١٣ مثلا
    int RandNum = rand() % (To - From + 1) + From;
    // 87513 % "10" ( 10 - 1 +1 ) = 3 + 1 = 4
    return RandNum; // 4
}

int main()
{
    srand((unsigned)time(NULL));
    // يعطي رقم عشوائي مع كل استدعاء

    cout << RandomNumber(1, 10) << endl;
    cout << RandomNumber(1, 10) << endl;
    cout << RandomNumber(1, 10) << endl;
}
```

حل المشكلة ٢٠

اكتب برنامجاً لطباعة حرف صغير عشوائي ، وحرف كبير عشوائي ، وحرف خاص – أو رمز – عشوائي ، وعدد عشوائي من ٠ الى ٩

المخرجات :

n

G

\$

7

م	Function & Procedures ...	ماذا يعمل ؟
١	int RandomNumber	رقم عشوائي
٢	enum enCharTayp	أنواع الحروف – الرموز –
٣	char GetRandomCharactea	الحصول على حرف عشوائي

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	00000000	NUL	32	0x20	040	01000000	space	64	0x40	100	10000000	@	96	0x60	140	11000000	`
1	0x01	001	00000001	SOH	33	0x21	041	01000001	!	65	0x41	101	10000001	A	97	0x61	141	11000001	a
2	0x02	002	00000010	STX	34	0x22	042	01000010	"	66	0x42	102	10000010	B	98	0x62	142	11000010	b
3	0x03	003	00000011	ETX	35	0x23	043	01000011	#	67	0x43	103	10000011	C	99	0x63	143	11000011	c
4	0x04	004	00000100	EOT	36	0x24	044	01000100	\$	68	0x44	104	10000100	D	100	0x64	144	11000100	d
5	0x05	005	00000101	ENQ	37	0x25	045	01000101	%	69	0x45	105	10000101	E	101	0x65	145	11000101	e
6	0x06	006	00000110	ACK	38	0x26	046	01000110	&	70	0x46	106	10000110	F	102	0x66	146	11000110	f
7	0x07	007	00000111	BEL	39	0x27	047	01000111	'	71	0x47	107	10000111	G	103	0x67	147	11000111	g
8	0x08	010	00010000	BS	40	0x28	050	01010000	{	72	0x48	110	10010000	H	104	0x68	150	11010000	h
9	0x09	011	00010001	TAB	41	0x29	051	01010001	}	73	0x49	111	10010001	I	105	0x69	151	11010001	i
10	0x0A	012	00010010	LF	42	0x2A	052	01010010	*	74	0x4A	112	10010010	J	106	0x6A	152	11010010	j
11	0x0B	013	00010011	VT	43	0x2B	053	01010011	+	75	0x4B	113	10010011	K	107	0x6B	153	11010011	k
12	0x0C	014	00010100	FF	44	0x2C	054	01010100	,	76	0x4C	114	10010100	L	108	0x6C	154	11010100	l
13	0x0D	015	00010101	CR	45	0x2D	055	01010101	-	77	0x4D	115	10010101	M	109	0x6D	155	11010101	m
14	0x0E	016	00010110	SO	46	0x2E	056	01010110	.	78	0x4E	116	10010110	N	110	0x6E	156	11010110	n
15	0x0F	017	00010111	SI	47	0x2F	057	01010111	/	79	0x4F	117	10010111	O	111	0x6F	157	11010111	o
16	0x10	020	00100000	DLE	48	0x30	060	01100000	0	80	0x50	120	10100000	P	112	0x70	160	11100000	p
17	0x11	021	00100001	DC1	49	0x31	061	01100001	1	81	0x51	121	10100001	Q	113	0x71	161	11100001	q
18	0x12	022	00100010	DC2	50	0x32	062	01100010	2	82	0x52	122	10100010	R	114	0x72	162	11100010	r
19	0x13	023	00100011	DC3	51	0x33	063	01100011	3	83	0x53	123	10100011	S	115	0x73	163	11100011	s
20	0x14	024	00100100	DC4	52	0x34	064	01100100	4	84	0x54	124	10100100	T	116	0x74	164	11100100	t
21	0x15	025	00100101	NAK	53	0x35	065	01100101	5	85	0x55	125	10100101	U	117	0x75	165	11100101	u
22	0x16	026	00100110	SYN	54	0x36	066	01100110	6	86	0x56	126	10100110	V	118	0x76	166	11100110	v
23	0x17	027	00100111	ETB	55	0x37	067	01100111	7	87	0x57	127	10100111	W	119	0x77	167	11100111	w
24	0x18	030	00110000	CAN	56	0x38	070	01110000	8	88	0x58	130	10110000	X	120	0x78	170	11110000	x
25	0x19	031	00110001	EM	57	0x39	071	01110001	9	89	0x59	131	10110001	Y	121	0x79	171	11110001	y
26	0x1A	032	00110010	SUB	58	0x3A	072	01110010	:	90	0x5A	132	10110010	Z	122	0x7A	172	11110010	z
27	0x1B	033	00110011	ESC	59	0x3B	073	01110011	;	91	0x5B	133	10110011	[123	0x7B	173	11110011	{
28	0x1C	034	00110100	FS	60	0x3C	074	01110100	<	92	0x5C	134	10110100	\	124	0x7C	174	11110100	
29	0x1D	035	00110101	GS	61	0x3D	075	01110101	=	93	0x5D	135	10110101]	125	0x7D	175	11110101	}
30	0x1E	036	00110110	RS	62	0x3E	076	01110110	>	94	0x5E	136	10110110	^	126	0x7E	176	11110110	~
31	0x1F	037	00110111	US	63	0x3F	077	01110111	?	95	0x5F	137	10110111	_	127	0x7F	177	11110111	DEL

جدول الأسكي كود


```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    // rand() int \٨٧٥١٣ = حجمه مثل = يعطي رقم عشوائي عادة يكون الرقم كبير مثلا
    int RandNum = rand() % (To - From + 1) + From;
    // 87513 % "25" ( 122 - 97 +1 ) = 13 + 97 = 110
    return RandNum; // 110
}

enum enCharTayp {SmallLetter = 1, CapitalLetter = 2 , SpecialCharacter = 3, Digit = 4};

char GetRandomCharacter(enCharTayp CharType)
{
    switch (CharType)
    {
        case enCharTayp::SmallLetter:
        {
            return char(RandomNumber(97, 122)); // char( 110 ) = n
            break;
        }

        case enCharTayp::CapitalLetter:
        {
            return char(RandomNumber(65, 90));
            break;
        }

        case enCharTayp::SpecialCharacter:
        {
            return char(RandomNumber(33, 47));
            break;
        }

        case enCharTayp::Digit:
        {
            return char(RandomNumber(48, 57));
            break;
        }
    }
}

int main()
{
    srand((unsigned)time(NULL));
    // يعطي رقم عشوائي مع كل استدعاء

    cout << GetRandomCharacter(enCharTayp::SmallLetter) << endl;
    cout << GetRandomCharacter(enCharTayp::CapitalLetter) << endl;
    cout << GetRandomCharacter(enCharTayp::SpecialCharacter) << endl;
    cout << GetRandomCharacter(enCharTayp::Digit) << endl;
}

```

حل المشكلة ٢١

اكتب برنامجاً لقراءة عدد المفاتيح المراد إنشاؤها وطباعتها على الشاشة

المخرجات :

Kay [1] : SDEK-TZCN-BUPN-TZUV

Kay [2] : OAFY-DNFT-KBKL-VZYR

Kay [3] : JSRL-PLMG-YWNV-CGDR

Kay [4] : ROTC-TKZG-XASN-BYPQ

Kay [5] : GGYB-DVHN-LZPJ-OTPF

ماذا يعمل ؟	Function & Procedures ...	م
رقم عشوائي	<code>int</code> RandomNumber	١
أنواع الحروف – الرموز –	<code>enum</code> enCharTayp	٢
الحصول على حرف عشوائي	<code>char</code> GetRandomCharactea	٣
عدد المفاتيح المراد إنشاؤها	<code>int</code> ReadPositiveNumber	٤
جمع الحرف مع حرف (العشوائية) لتكوين كلمة	<code>string</code> GenerateWord	٥
جمع كلمة مع كلمة (العشوائية) لتكوين كلمة السر	<code>string</code> GenerateKey	٦
تكرار عدد المفاتيح مع كلمة السر	<code>void</code> GenerateKeys	٧

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    // rand() int \٨٧١٣ مثل = يعطي رقم عشوائي عادة يكون الرقم كبير مثلا
    int RandNum = rand() % (To - From + 1) + From;
    // 87513 % "25" ( 122 - 97 +1 ) = 13 + 97 = 110
    return RandNum; // 110
}

enum enCharTayp { SmallLetter = 1, CapitalLetter = 2, SpecialCharacter = 3, Digit = 4 };

char GetRandomCharacter(enCharTayp CharType)
{
    switch (CharType)
    {
        case enCharTayp::SmallLetter:
        {
            return char(RandomNumber(97, 122)); // char( 110 ) = n
            break;
        }

        case enCharTayp::CapitalLetter:
        {
            return char(RandomNumber(65, 90));
            break;
        }

        case enCharTayp::SpecialCharacter:
        {
            return char(RandomNumber(33, 47));
            break;
        }

        case enCharTayp::Digit:
        {
            return char(RandomNumber(48, 57));
            break;
        }
    }
}

int ReadPositiveNumber(string Message)
{
    int Number;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number < 0);

    return Number;
}

```

```

string GenerateWord(enCharTayp CharType, short Length)
{
    string Word;

    for (int i = 1; i <= Length; i++) // 1 <= 4 ? | 2 <= 4 ...
    {
        Word += GetRandomCharacter(CharType);
    }
    return Word;
}

string GenerateKey()
{
    string Key = "";

    Key = GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4);

    return Key;
}

void GenerateKeys(short NumberOfKeys)
{
    for (int i = 1; i <= NumberOfKeys; i++)
    {
        cout << "Kay [" << i << "]" : " << GenerateKey() << endl;
    }
}

int main()
{
    srand((unsigned)time(NULL));
    // يعطي رقم عشوائي مع كل استدعاء

    GenerateKeys(ReadPositiveNumber("Pleas enter how many keys to
generate ?"));
}

```

حل المشكلة ٢٢

اكتب برنامجا لقراءة عناصر وتخزينها في مصفوفة ، ثم اطبع جميع عناصر المصفوفة ،
واطلب رقما للتحقق منه ، ثم اطبع العناصر المكررة في عنصر معين

المدخلات :

Enter number of elements : 5

Elements [1] : 1

Elements [2] : 1

Elements [3] : 1

Elements [4] : 2

Elements [5] : 3

: Enter the number you want to check

1

المخرجات :

Original array : 1 1 1 2 3

.Number 1 is repeated 3 time(s)

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> ReadPositiveNumber	رقم للتحقق منه في العناصر
٢	<code>void</code> ReadArray	قراء عناصر المصفوفة
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة
٤	<code>int</code> TimeRepeated	طباعة كم مرة تكرر العنصر المعين

```
#include <iostream>
using namespace std;
int ReadPositiveNumber(string Message)
{
    int Number;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number < 0);

    return Number;
}
```

```

void ReadArray(int arr[100], int& arrLength)
// & main (تخزن) القيمة المدخلة في
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;
    cout << "\n";

    for (int i = 0 ; i < arrLength; i++)
    // مكان تخزين أول عنصر في arr
    {
        cout << "Elements [" << i +1 << "]" : ";
    // [1] == arr أول قيمة عنصر في
        cin >> arr[i];
    }
    cout << endl;
}

void PrintArray(int arr[100], int arrLength)
// main استدعاء القيمة المخزنة في arrLength
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int TimeRepeated(int Number, int arr[100], int arrLength)
{
    int Count = 0;
    for (int i = 0; i <= arrLength - 1; i++)
    {
        if (Number == arr[i])
            Count++;
    }
    return Count;
}

int main()
{
    int arr[100], arrLength, NumberToCheck;

    ReadArray(arr, arrLength);

    NumberToCheck = ReadPositiveNumber("Enter the number you want to
check : ");

    cout << "\nOriginal array : ";
    PrintArray(arr, arrLength);

    cout << "\nNumber " << NumberToCheck;
    cout << " is repeated ";

    cout << TimeRepeated(NumberToCheck, arr, arrLength) << "
time(s).\n";
}

```

حل المشكلة ٢٣

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠

المدخلات :

Enter number of elements : 5

المخرجات :

Original array : 12 34 67 97 56

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية

```
#include <iostream>
#include <cstdlib>
using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
// & main (تخزن ) القيمة المدخلة في
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
// index[0] == arr مكان تخزين أول قيمة في
    arr[i] = RandomNumber(1, 100); // [1] == arr أول قيمة عنصر في
}

void PrintArray(int arr[100], int arrLength)
// main استدعاء القيمة المخزنة في
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int main()
{
    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : ";
    PrintArray(arr, arrLength);
}
```

حل المشكلة ٢٤

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم اطبع أكبر رقم

المدخلات :

Enter number of elements : 5

المخرجات :

Array Elements : 12 34 67 97 56

Max Number is : 97

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> MaxNumberInArray	طباعة أكبر رقم عشوائي


```

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

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength) // & main
    لترجع ( تخزن ) القيمة المدخلة في
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++) // index[0] == arr مكان تخزين
        أول قيمة في
        arr[i] = RandomNumber(1, 100); // [1] == arr أول قيمة عنصر
    في
}

void PrintArray(int arr[100], int arrLength) // main استدعاء القيمة
    arrLength المخزنة في
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int MaxNumberInArray(int arr[100], int arrLength)
{
    int Max = 0;
    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] > Max)
        {
            Max = arr[i];
        }
    }
    return Max;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : ";
    PrintArray(arr, arrLength);

    cout << "\nMax Number is : " <<
        MaxNumberInArray(arr, arrLength) << endl;
}

```

حل المشكلة ٢٥

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم اطبع أصغر رقم

المدخلات :

Enter number of elements : 5

المخرجات :

Array Elements : 12 34 67 97 56

Min Number is : 12

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> MinNumberInArray	طباعة أصغر رقم عشوائي

```

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

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
// & main (تخزن) القيمة المدخلة في
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
    // index[0] == arr مكان تخزين أول قيمة في arr
    arr[i] = RandomNumber(1, 100); // [1] == arr أول قيمة عنصر في
}

void PrintArray(int arr[100], int arrLength)
// main استدعاء القيمة المخزنة في arrLength
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int MinNumberInArray(int arr[100], int arrLength)
{
    int Min = 0;
    Min = arr[0];

    for (int i = 1; i < arrLength; i++)
    {
        if (arr[i] < Min)
        {
            Min = arr[i];
        }
    }
    return Min;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : ";
    PrintArray(arr, arrLength);

    cout << "\nMin Number is : " <<
        MinNumberInArray(arr, arrLength) << endl;
}

```

حل المشكلة ٢٦

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم اطبع مجموع الأرقام

المدخلات :

Enter number of elements : 5

المخرجات :

Array Elements : 12 34 67 97 56

Sum Of all Number is : 266

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> SumArray	طباعة مجموع الأرقام العشوائية

```

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

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
// & main (تخزن) القيمة المدخلة في
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
    // index[0] == arr مكان تخزين أول قيمة في
        arr[i] = RandomNumber(1, 100); // [1] == arr أول قيمة عنصر في
}

void PrintArray(int arr[100], int arrLength)
// main استدعاء القيمة المخزنة في
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int SumArray(int arr[100], int arrLength)
{
    int Sum = 0;

    for (int i = 1; i < arrLength; i++)
    {
        Sum += arr[i];
    }
    return Sum;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : ";
    PrintArray(arr, arrLength);

    cout << "\nSum of all Number is : " <<
        SumArray(arr, arrLength) << endl;
}

```

حل المشكلة ٢٧

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم اطبع المتوسط الحسابي للأرقام

المدخلات :

Enter number of elements : 5

المخرجات :

Array Elements : 12 34 67 97 56

Average Of all Number is : 53.2

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> SumArray	مجموع الأرقام العشوائية
٥	<code>float</code> ArrayAverage	طباعة المتوسط الحسابي للأرقام العشوائية

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int SumArray(int arr[100], int arrLength)
{
    int Sum = 0;

    for (int i = 1; i < arrLength; i++)
    {
        Sum += arr[i];
    }
    return Sum;
}

float ArrayAverage(int arr[100] , int arrLength )
{
    return (float) SumArray(arr , arrLength) / arrLength;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : ";
    PrintArray(arr, arrLength);

    cout << "\nAverage of all Number is : ";
    cout << ArrayAverage(arr, arrLength) << endl;
}

```

حل المشكلة ٢٨

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم انسخها الى مصفوفة أخرى واطبعها

المدخلات :

Enter number of elements : 5

المخرجات :

Array 1 Elements : 12 34 67 97 56

Array 2 Elements after copy : 12 34 67 97 56

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>void</code> CopyArray	نسخ عناصر المصفوفة للأرقام العشوائية


```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void CopyArray(int arrSource[100], int arrDestination[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        arrDestination[i] = arrSource[i];
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    int arr2[100];
    CopyArray(arr, arr2, arrLength); // arr == arr2

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Elements after copy : \n";
    PrintArray(arr2, arrLength);
}

```

حل المشكلة ٢٩

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم انسخ الأعداد الأولية فقط الى مصفوفة أخرى واطبعها

المدخلات :

Enter number of elements : 5

المخرجات :

Array 1 Elements : 12 34 67 97 56

Prime Number in Array 2 : 67 97

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>enum enPrimNotPrime</code>	تخزين قيمة (عدد أولي ، عدد غير أولي)
٢	<code>enPrimNotPrime CheckPrime</code>	اختيار الأعداد (الأولية ، غير الأولية)
٣	<code>int RandomNumber</code>	رقم عشوائي
٤	<code>void FillArrayWithRandomNumbers</code>	تعبئة عناصر المصفوفة لأرقام عشوائية
٥	<code>void PrintArray</code>	طباعة عناصر المصفوفة للأرقام العشوائية
٦	<code>void CopyOnlyPrimeNumbers</code>	اختيار الأرقام الأولية من الأرقام العشوائية

```
#include <iostream>
#include <cstdlib>

using namespace std;

enum enPrimNotPrime { Prime = 1, NotPrime = 2 };

enPrimNotPrime CheckPrime(int Number)
{
    int M = round(Number / 2);

    for (int Counter = 2; Counter <= M; Counter++)
    {
        if (Number % Counter == 0)
            return enPrimNotPrime::NotPrime;
    }
    return enPrimNotPrime::Prime;;
}

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}
```

```

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0 ; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void CopyOnlyPrimeNumbers(int arrSource[100], int arrDestination[100], int
arrLength, int& arr2Length)
{
    int Counter = 0;

    for (int i = 0; i < arrLength; i++)
    {
        if (CheckPrime(arrSource[i]) == enPrimNotPrime::Prime)
        {
            arrDestination[Counter] = arrSource[i];
            Counter++;
        }
    }
    arr2Length = --Counter; // لأن العداد كان زائد واحد
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    int arr2[100] , arr2Length = 0;
    CopyOnlyPrimeNumbers(arr, arr2, arrLength , arr2Length);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nPrime Numbers in Array 2 : \n";
    PrintArray(arr2, arr2Length);
}

```

حل المشكلة ٣٠

اكتب برنامجا لملء مصفوفتين بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ وجمع عناصرها في مصفوفة ثالثة وطباعتها

المدخلات :

Enter number of elements : 5

المخرجات :

Array 1 Elements : 12 34 67 97 56

Array 2 Elements : 56 87 34 98 8

Sum of array 1 and array 2 elements : 68 121 101 195 64

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم (عدد العناصر)	<code>int ReadPositiveNumber</code>	١
رقم عشوائي	<code>int RandomNumber</code>	٢
تعبئة عناصر المصفوفة لأرقام عشوائية	<code>void FillArrayWithRandomNumbers</code>	٣
جمع عناصر المصفوفة ١ والمصفوفة ٢	<code>void SumOf2Arrays</code>	٤
طباعة عناصر المصفوفة للأرقام العشوائية	<code>void PrintArray</code>	٥

```
#include <iostream>
#include <cstdlib>

using namespace std;

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number <= 0);

    return Number;
}

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}
```

```

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    for (int i = 0 ; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void SumOf2Arrays(int arr[100], int arr2[100], int arrSum[100], int
arrLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        arrSum[i] = arr[i] + arr2[i];
    }
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100] ,arr2[100], arrSum[100];
    int arrLength = ReadPositiveNumber("Enter number of elements : ");

    FillArrayWithRandomNumbers(arr, arrLength);
    FillArrayWithRandomNumbers(arr2, arrLength);

    SumOf2Arrays(arr, arr2 , arrSum , arrLength );

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Elements : \n";
    PrintArray(arr2, arrLength);

    cout << "\nSum of array 1 and array 2 elements : \n";
    PrintArray(arrSum, arrLength);
}

```

حل المشكلة ٣١

اكتب برنامجا لملء المصفوفة بأرقام مرتبة من ١ الى الرقم، ثم اطبعها ، ثم قم بخلط هذه المصفوفة وطباعتها بعد خلطها

المدخلات :

Enter number of elements : 5

المخرجات :

Array elements before shuffle:: 1 2 3 4 5

Array elements after shuffle:: 3 2 5 1 4

ماذا يعمل ؟	Function & Procedures ...	م
قراءة رقم (عدد العناصر)	<code>int</code> ReadPositiveNumber	١
تبديل عنصر ١ ب عنصر ٢ والعكس	<code>void</code> Swap	٢
طباعة الأرقام من ١ الى الرقم المدخل (عدد العناصر)	<code>void</code> FillArrayWith1ToNum	٣
رقم عشوائي	<code>int</code> RandomNumber	٤
خلط العناصر بشكل عشوائي	<code>void</code> ShuffleArray	٥
طباعة عناصر المصفوفة للأرقام العشوائية	<code>void</code> PrintArray	٦

```
#include <iostream>
#include <cstdlib>

using namespace std;

int ReadPositiveNumber(string Message)
{
    int Number = 0;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number <= 0);

    return Number;
}

void Swap(int& A , int& B)
{
    int Temp;

    Temp = A;
    A = B;
    B = Temp;
}
```

```

void FillArrayWith1ToNum(int arr[100], int arrLength)
{
    for (int i = 0; i <= arrLength; i++)
    {
        arr[i] = i + 1;
    }
}

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void ShuffleArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        Swap(arr[RandomNumber(1, arrLength) - 1], arr[RandomNumber(1,
arrLength) - 1]);
    }
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100];
    int arrLength = ReadPositiveNumber("Enter number of elements : ");

    FillArrayWith1ToNum(arr, arrLength);

    cout << "\nArray elements before shuffle: \n";
    PrintArray(arr, arrLength);

    ShuffleArray(arr, arrLength);

    cout << "\nArray elements after shuffle: \n";
    PrintArray(arr, arrLength);
}

```

حل المشكلة ٣٢

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ثم انسخها الى مصفوفة أخرى واطبعها

المدخلات :

Enter number of elements : 5

المخرجات :

Array 1 Elements : 12 34 67 97 56

Array 2 Elements after copy : 56 97 67 34 12

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>void</code> CopyArrayInReverseOrder	طباعة عناصر المصفوفة بشكل معكوس


```

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

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void CopyArrayInReverseOrder(int arrSource[100], int arrDestination[100],
int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        arrDestination[i] = arrSource[arrLength-1-i];
    // arrDestination[0] = arrSource[5] | arrDestination[1] = arrSource[4]
    ....
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;

    FillArrayWithRandomNumbers(arr, arrLength);

    int arr2[100];
    CopyArrayInReverseOrder(arr, arr2, arrLength);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Elements after copy : \n";
    PrintArray(arr2, arrLength);
}

```

حل المشكلة ٣٣

اكتب برنامجا لقراءة عدد المفاتيح المراد إنشاؤها وتعبئتها في مصفوفة ثم طباعتها على الشاشة المدخلات :

How many keys do you want to generate ?

5

المخرجات :

: Array elements

Array [0] : MKJM-ZEKN-QJDK-CQOP

Array [1] : PQDV-XRGW-QXNW-LWJA

Array [2] : EQKK-CIDQ-XOJE-BHJH

Array [3] : XATF-XPNG-CJMC-PISC

Array [4] : DHXG-DISJ-YXUE-NJLL

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>enum</code> enCharTayp	أنواع الحروف – الرموز –
٣	<code>char</code> GetRandomCharactea	الحصول على حرف عشوائي
٤	<code>int</code> ReadPositiveNumber	عدد المفاتيح المراد إنشاؤها
٥	<code>string</code> GenerateWord	جمع الحرف مع حرف (العشوائية) لتكوين كلمة
٦	<code>string</code> GenerateKey	جمع كلمة مع كلمة (العشوائية) لتكوين كلمة السر
٧	<code>void</code> FillArrayWithKeys	تكرار عدد المفاتيح مع كلمة السر
٨	<code>void</code> PrintStringArray	طباعة المصفوفة مع كلمة السر

```

#include <iostream>
#include <cstdlib>
using namespace std;
int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum; // 110
}
enum enCharTayp { SmallLetter = 1, CapitalLetter = 2, SpecialCharacter =
3, Digit = 4 };

char GetRandomCharacter(enCharTayp CharType)
{
    switch (CharType)
    {
        case enCharTayp::SmallLetter:
        {
            return char(RandomNumber(97, 122));
            break;
        }

        case enCharTayp::CapitalLetter:
        {
            return char(RandomNumber(65, 90));
            break;
        }

        case enCharTayp::SpecialCharacter:
        {
            return char(RandomNumber(33, 47));
            break;
        }

        case enCharTayp::Digit:
        {
            return char(RandomNumber(48, 57));
            break;
        }
    }
}

void PrintStringArray(string arr[100], int arrLength)
{
    cout << "\nArray elements : \n\n";
    for (int i = 0; i < arrLength; i++)
    {
        cout << "Array [" << i << "]" : ";
        cout << arr[i] << "\n";
    }
    cout << endl;
}

int ReadPositiveNumber(string Message)
{
    int Number;
    do
    {
        cout << Message << endl;
        cin >> Number;
    } while (Number < 0);

    return Number;
}

```

```

string GenerateWord(enCharTayp CharType, short Length)
{
    string Word;

    for (int i = 1; i <= Length; i++) // 1 <= 4 ? | 2 <= 4 ...
    {
        Word += GetRandomCharacter(CharType);
    }
    return Word;
}

string GenerateKey()
{
    string Key = "";

    Key = GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4) + "-";
    Key = Key + GenerateWord(enCharTayp::CapitalLetter, 4);

    return Key;
}

void FillArrayWithKeys(string arr[100] , int arrLength)
{
    for (int i = 0; i < arrLength ; i++)
    {
        arr[i] = GenerateKey();
    }
}

int main()
{
    srand((unsigned)time(NULL));
    // يعطي رقم عشوائي مع كل استدعاء

    string arr[100];
    int arrLength = 0;
    arrLength = ReadPositiveNumber("How many keys do you want to
generate ?");

    FillArrayWithKeys(arr, arrLength);
    PrintStringArray(arr, arrLength);
}

```

حل المشكلة ٣٤

اكتب برنامجاً لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، واقرأ رقم وأعد موقعه (index) ، وإذا وجد خلاف ذلك أعد -١

المدخلات :

: Enter number of elements

5

المخرجات :

: Array 1 Elements

33 9 55 42 82

Please enter a number to search for?

42

Number you are looking for is : 42

The number found at position : 3

The number found its order : 4

أو

Please enter a number to search for?

3

Number you are looking for is : 3

The number is not found :-)

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int RandomNumber</code>	رقم عشوائي
٢	<code>void FillArrayWithRandomNumbers</code>	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void PrintArray</code>	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>short FindNumberPositionInArray</code>	البحث عن الرقم وإرجاع قيمته
٥	<code>int ReadNumber</code>	قراءة رقم للبحث عنه داخل المصفوفة

```
#include <iostream>
#include <cstdlib>
using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}
```

```

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}
void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}
short FindNumberPositionInArray(int Number, int arr[100], int arrLength)
{
    // هذا الفانكشن يبحث عن الرقم في المصفوفة ويعيد الإنديكس إذا وجد
    // وإذا لم يجد الرقم يعيد -١ أي لم يتم العثور على الرقم
    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] == Number)
            return i;
    }
    return -1;
}
int ReadNumber()
{
    int Number;
    cout << "\nPlease enter a number to search for?\n";
    cin >> Number;
    return Number;
}
int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;
    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    int Number = ReadNumber();
    cout << "\nNumber you are looking for is : " << Number << endl;

    short NumberPosition = FindNumberPositionInArray(Number, arr,
arrLength);

    if (NumberPosition == -1)
    {
        cout << "\nThe number is not found :-( \n";
    }
    else
    {
        cout << "The number found at position : " << NumberPosition
<< endl;
        cout << "The number found its order : " << NumberPosition +1
<< endl;
    }
}

```

حل المشكلة ٣٥

اكتب برنامجاً لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، واقرأ الرقم واطبع إذا وجده أم لا ، (أعد استخدام الكود في المشكلة السابقة)

المدخلات :

Enter number of elements :

5

المخرجات :

: Array 1 Elements

51 60 71 19 83

Please enter a number to search for?

71

Number you are looking for is : 71

Yes , it is found :-)

أو

Please enter a number to search for?

2

Number you are looking for is : 2

No , The number is not found :-)

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>short</code> FindNumberPositionInArray	البحث عن الرقم وإرجاع قيمته
٥	<code>int</code> ReadNumber	قراءة رقم للبحث عنه داخل المصفوفة
٦	<code>bool</code> IsNumberInArray	هل الرقم موجود في المصفوفة ؟ نعم أو لا

```
#include <iostream>
#include <cstdlib>
using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}
```

```

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

short FindNumberPositionInArray(int Number, int arr[100], int arrLength)
{
    // هذا الفانكشن يبحث عن الرقم في المصفوفة ويعيد الإنديكس إذا وجد الرقم
    // وإذا لم يجد الرقم يعيد -1 أي لم يتم العثور على الرقم

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] == Number)
            return i;
    }
    return -1;
}

int ReadNumber()
{
    int Number;
    cout << "\nPlease enter a number to search for?\n";
    cin >> Number;
    return Number;
}

bool IsNumberInArray(int Number, int arr[100], int arrLength)
{
    return FindNumberPositionInArray( Number, arr , arrLength) != -1;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength;
    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    int Number = ReadNumber();
    cout << "\nNumber you are looking for is : " << Number << endl;

    if (! IsNumberInArray(Number, arr, arrLength))
    {
        cout << "No , The number is not found :-( \n";
    }
    else
    {
        cout << "Yes , it is found :-)" << endl;
    }
}

```


حل المشكلة ٣٦

اكتب برنامجا لقراءة الأرقام ديناميكيا وحفظها في مصفوفة , الحد الأقصى لحجم المصفوفة ١٠٠ ، قم بتخصيص طول المصفوفة ديناميكيا

المخرجات :

Please enter a number? 10

Do you want to add more numbers? [0]:No,[1]:yes? 1

Please enter a number? 20

Do you want to add more numbers? [0]:No,[1]:yes? 1

Please enter a number? 30

Do you want to add more numbers? [0]:No,[1]:yes? 0

Array Length: 3

Array elements: 10 20 30

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> ReadNumber	قراءة رقم
٢	<code>void</code> AddArrayElement	إضافة طول المصفوفة
٣	<code>void</code> InputUserNumbersInArray(إنشاء مصفوفة جديدة (اسأل المستخدم)
٤	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية

```

#include<iostream>
using namespace std;

int ReadNumber()
{
    int Number;
    cout << "\nPlease enter a number? ";
    cin >> Number;

    return Number;
}

void AddArrayElement(int Number, int arr[100], int& arrLength)
{ //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void InputUserNumbersInArray(int arr[100], int& arrLength)
{
    bool AddMore = true;
    do
    {
        AddArrayElement(ReadNumber(), arr, arrLength);
        cout << "\nDo you want to add more numbers? [0]:No,[1]:yes?
";
        cin >> AddMore;
    } while (AddMore);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int main()
{
    int arr[100], arrLength = 0; // وضع الصفر مهم

    InputUserNumbersInArray(arr, arrLength);

    cout << "\nArray Length: " << arrLength << endl;
    cout << "Array elements: ";
    PrintArray(arr, arrLength);

    return 0;
}

```

حل المشكلة ٣٧

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، وانسخها الى مصفوفة أخرى باستخدام AddArrayElement ثم اطبعها

المدخلات :

Enter number of elements :

5

المخرجات :

: Array 1 Elements

82 37 50 62 25

: Array 2 Elements after copy

82 37 50 62 25

م	Function & Procedures ...	ماذا يعمل ؟
١	int RandomNumber	رقم عشوائي
٢	void FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	void PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	void AddArrayElement	إضافة طول المصفوفة
٥	void CopyArrayUsingAddArrayElement	نسخ القيم الى المصفوفة

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void AddArrayElement(int Number, int arr[100], int& arrLength)
{ //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void CopyArrayUsingAddArrayElement(int arrSource[100], int
arrDestination[100], int arrLength , int& arrDestinationLength)
{
    for (int i = 0; i < arrLength; i++)
        AddArrayElement(arrSource[i] , arrDestination ,
arrDestinationLength);
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0, arr2Length = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    int arr2[100];
    CopyArrayUsingAddArrayElement(arr, arr2, arrLength , arr2Length);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Odd number : \n";
    PrintArray(arr2, arrLength);
}

```

}

حل المشكلة ٣٨

اكتب برنامجاً لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، وانسخ الأرقام الفردية فقط الى مصفوفة أخرى باستخدام AddArrayElement ثم اطبعها

المدخلات :

Enter number of elements :

5

المخرجات :

: Array 1 Elements

82 37 50 62 25

: Array 2 Elements after copy

37 25

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>void</code> AddArrayElement	إضافة طول المصفوفة
٥	<code>void</code> CopyOddNumbers	نسخ القيم الفردية فقط الى المصفوفة

```
#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}
```

```

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void AddArrayElement(int Number, int arr[100], int& arrLength)
{ //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void CopyOddNumbers(int arrSource[100], int arrDestination[100], int
arrLength, int& arrDestinationLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        if (arrSource[i] % 2 != 0)
        {
            AddArrayElement(arrSource[i], arrDestination,
arrDestinationLength);
        }
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0, arr2Length = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    int arr2[100];
    CopyOddNumbers(arr, arr2, arrLength , arr2Length);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Elements after copy : \n";
    PrintArray(arr2, arr2Length);
}

```

حل المشكلة ٣٩

اكتب برنامجاً لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، وانسخ الأعداد الأولية فقط الى مصفوفة أخرى باستخدام AddArrayElement ثم اطبعها

المدخلات :

Enter number of elements :

5

المخرجات :

: Array 1 Elements

82 37 50 62 25

: Array 2 Elements after copy

37

ماذا يعمل ؟	Function & Procedures ...	م
تخزين القيم	enum enPrimNotPrime	١
اختيار العدد الأولي	enPrimNotPrime CheckPrime	٢
رقم عشوائي	int RandomNumber	٣
تعبئة عناصر المصفوفة لأرقام عشوائية	void FillArrayWithRandomNumbers	٤
طباعة عناصر المصفوفة للأرقام العشوائية	void PrintArray	٥
إضافة طول المصفوفة	void AddArrayElement	٦
طباعة الأعداد الأولية	void CopyPrimeNumbers	٧

```
#include <iostream>
#include <cstdlib>

using namespace std;

enum enPrimNotPrime { Prime = 1, NotPrime = 2 };

enPrimNotPrime CheckPrime(int Number)
{
    int M = round(Number / 2);

    for (int Counter = 2; Counter <= M; Counter++)
    {
        if (Number % Counter == 0)
            return enPrimNotPrime::NotPrime;
    }
    return enPrimNotPrime::Prime;
}
```

```

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

void AddArrayElement(int Number, int arr[100], int& arrLength)
{ //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void CopyPrimeNumbers(int arrSource[100], int arrDestination[100], int
arrLength, int& arrDestinationLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        if (CheckPrime(arrSource[i]) == enPrimNotPrime::Prime)
        {
            AddArrayElement(arrSource[i], arrDestination,
arrDestinationLength);
        }
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0, arr2Length = 0;
    int arr2[100];

    FillArrayWithRandomNumbers(arr, arrLength);

    CopyPrimeNumbers(arr, arr2, arrLength , arr2Length);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nArray 2 Prime Number : \n";
    PrintArray(arr2, arr2Length);
}

```


حل المشكلة ٤٠

اكتب برنامجا لملء مصفوفة بالأرقام ، ثم اطبع الأرقام الفريدة من نوعها

المخرجات :

: Array 1 Elements

10 10 10 50 50 70 70 70 70 90

: Array 1 distinct Elements

10 50 70 90

ماذا يعمل ؟	Function & Procedures ...	م
تخزين قيم في المصفوفة	<code>void FillArray</code>	١
طباعة عناصر المصفوفة	<code>void PrintArray</code>	٢
البحث عن رقم وإرجاع موقعه	<code>short FindNumberPositionInArray</code>	٣
هل الرقم متكرر في المصفوفة ؟ نعم أو لا	<code>bool IsNumberInArray</code>	٤
إضافة طول المصفوفة	<code>void AddArrayElement</code>	٥
طباعة الأعداد الفريدة من نوعها	<code>void CopyDistinactNumbersToArray</code>	٦

```
#include <iostream>

using namespace std;

void FillArray(int arr[100], int& arrLength)
{
    arrLength = 10;

    arr[0] = 10;
    arr[1] = 10;
    arr[2] = 10;
    arr[3] = 50;
    arr[4] = 50;
    arr[5] = 70;
    arr[6] = 70;
    arr[7] = 70;
    arr[8] = 70;
    arr[9] = 90;
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}
```

```

short FindNumberPositionInArray(int Number, int arr[100], int arrLength)
{
    // هذا الفانكشن يبحث عن الرقم في المصفوفة ويعيد الإنديكس إذا وجد الرقم
    // وإذا لم يجد الرقم يعيد -١ أي لم يتم العثور على الرقم

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] == Number)
            return i;
    }
    return -1;
}

bool IsNumberInArray(int Number, int arr[100], int arrLength)
{
    return FindNumberPositionInArray(Number, arr, arrLength) != -1;
}

void AddArrayElement(int Number, int arr[100], int& arrLength)
{ //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void CopyDistinactNumbersToArray(int arrSource[100], int
arrDestination[100], int arrLength, int& arrDestinationLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        if(! IsNumberInArray(arrSource[i],arrDestination,arrLength ))
        {
            AddArrayElement(arrSource[i], arrDestination,
arrDestinationLength);
        }
    }
}

int main()
{
    int arrSource[100], SourceLength = 0, arrDestination[100],
DestinationLength = 0;

    FillArray(arrSource, SourceLength);

    cout << "\nArray 1 Elements : \n";
    PrintArray(arrSource, SourceLength);

    CopyDistinactNumbersToArray(arrSource, arrDestination, SourceLength,
DestinationLength);

    cout << "\nArray 1 distivct Elements : \n";
    PrintArray(arrDestination, DestinationLength);
}

```

حل المشكلة ٤١

اكتب برنامجا لملء مصفوفة بالأرقام ، ثم تحقق مما إذا كانت المصفوفة يمكن قراءتها من اليمين الى اليسار ومن اليسار الى اليمين بنفس الطريقة (Palindrome)

المخرجات :

: Array Elements

10 20 30 30 20 10

Yes array is palindrome

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>void FillArray</code>	تخزين قيم في المصفوفة
٢	<code>void PrintArray</code>	طباعة عناصر المصفوفة
٣	<code>bool IsPalindromeArray</code>	هل يمكن قراءة الرقم من اليمين الى اليسار والعكس بنفس الطريقة ؟

```

#include <iostream>

using namespace std;

void FillArray(int arr[100], int& arrLength)
{
    arrLength = 6;

    arr[0] = 10;
    arr[1] = 20;
    arr[2] = 30;
    arr[3] = 30;
    arr[4] = 20;
    arr[5] = 10;
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

bool IsPalindromeArray( int arr[100], int Length)
{
    for (int i = 0; i < Length; i++)
    {
        if (arr[i] != arr[Length - i - 1])
        {
            return false;
        }
    }
    return true;
}

int main()
{
    int arr[100], Length = 0;

    FillArray(arr, Length);

    cout << "\nArray Elements : \n";
    PrintArray(arr, Length);

    if (IsPalindromeArray(arr, Length))
    {
        cout << "\nYes array is palindrome \n";
    }
    else
    {
        cout << "\nNo array is not palindrome \n";
    }
}

```

حل المشكلة ٤٢

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، ثم اطبع عدد الأرقام الفردية

المدخلات :

Enter number of elements :

5

المخرجات :

Array Elements :

87 66 30 87 70

: Odd numbers count is

2

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> OddCount	عد الأرقام الفردية

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int OddCount(int arr[100], int arrLength)
{
    int counter = 0;

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] % 2 != 0)
        {
            counter++;
        }
    }
    return counter;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nOdd numbers count is : \n";
    cout << OddCount(arr, arrLength) << endl;
}

```

حل المشكلة ٤٣

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١ الى ١٠٠ ، ثم اطبع عدد الأرقام الزوجية

المدخلات :

Enter number of elements :

5

المخرجات :

Array Elements :

43 27 4 94 45

Even numbers count is :

2

م	Function & Procedures ...	ماذا يعمل ؟
١	<code>int</code> RandomNumber	رقم عشوائي
٢	<code>void</code> FillArrayWithRandomNumbers	تعبئة عناصر المصفوفة لأرقام عشوائية
٣	<code>void</code> PrintArray	طباعة عناصر المصفوفة للأرقام العشوائية
٤	<code>int</code> EvenCount	عد الأرقام الزوجية

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(1, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int EvenCount(int arr[100], int arrLength)
{
    int counter = 0;

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] % 2 == 0)
        {
            counter++;
        }
    }
    return counter;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nEven numbers count is : \n";
    cout << EvenCount(arr, arrLength) << endl;
}

```


حل المشكلة ٤٤

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١٠٠ - الى ١٠٠ ، ثم
اطبع عدد الأرقام الموجبة

المدخلات :

Enter number of elements :

5

المخرجات :

Array Elements :

-11 27 54 -21 53

Positive numbers count is :

3

ماذا يعمل ؟	Function & Procedures ...	م
رقم عشوائي	<code>int</code> RandomNumber	١
تعبئة عناصر المصفوفة لأرقام عشوائية	<code>void</code> FillArrayWithRandomNumbers	٢
طباعة عناصر المصفوفة للأرقام العشوائية	<code>void</code> PrintArray	٣
عد الأرقام الموجبة	<code>int</code> PositiveCount	٤

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(-100, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int PositiveCount(int arr[100], int arrLength)
{
    int counter = 0;

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] >= 0)
        {
            counter++;
        }
    }
    return counter;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nPositive numbers count is : \n";
    cout << PositiveCount(arr, arrLength) << endl;
}

```

حل المشكلة ٤٥

اكتب برنامجا لملء مصفوفة بحجم أقصى ١٠٠ ، بأرقام عشوائية من ١٠٠ - الى ١٠٠ ، ثم
اطبع عدد الأرقام السالبة

المدخلات :

Enter number of elements :

5

المخرجات :

Array Elements :

-11 27 54 -21 53

Negative numbers count is :

2

ماذا يعمل ؟	Function & Procedures ...	م
رقم عشوائي	<code>int</code> RandomNumber	١
تعبئة عناصر المصفوفة لأرقام عشوائية	<code>void</code> FillArrayWithRandomNumbers	٢
طباعة عناصر المصفوفة للأرقام العشوائية	<code>void</code> PrintArray	٣
عد الأرقام السالبة	<code>int</code> NegativeCount	٤

```

#include <iostream>
#include <cstdlib>

using namespace std;

int RandomNumber(int From, int To)
{
    int RandNum = rand() % (To - From + 1) + From;
    return RandNum;
}

void FillArrayWithRandomNumbers(int arr[100], int& arrLength)
{
    cout << "Enter number of elements : \n";
    cin >> arrLength;

    for (int i = 0; i < arrLength; i++)
        arr[i] = RandomNumber(-100, 100);
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

int NegativeCount(int arr[100], int arrLength)
{
    int counter = 0;

    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] < 0)
        {
            counter++;
        }
    }
    return counter;
}

int main()
{
    srand((unsigned)time(NULL));

    int arr[100], arrLength = 0;

    FillArrayWithRandomNumbers(arr, arrLength);

    cout << "\nArray Elements : \n";
    PrintArray(arr, arrLength);

    cout << "\nNegative numbers count is : \n";
    cout << NegativeCount(arr, arrLength) << endl;
}

```

حل المشكلة ٤٦

اكتب برنامجا لطباعة القيمة المطلقة ، (لا تستخدم وظيفة جاهزة)

المدخلات :

Pleas enter a number ?

-10

المخرجات :

My abs Result : 10

C++ abs Result : 10

ماذا يعمل ؟	Function & Procedures ...	م
إرجاع القيمة المطلقة (الموجبة)	<code>float</code> MyABS	١
قراءة رقم	<code>float</code> ReadNumber	٢

```
#include <iostream>
#include <cmath>

using namespace std;

float MyABS(float Number)
{
    if (Number > 0)
        return Number;
    else
        return Number * -1;
}

float ReadNumber()
{
    float Number;
    cout << "Pleas enter a number ? " << endl;
    cin >> Number;

    return Number;
}

int main()
{
    float Number = ReadNumber();

    cout << "My abs Result : " << MyABS(Number) << endl;
    cout << "C++ abs Result : " << abs(Number) << endl;

}
```

حل المشكلة ٤٧

اكتب برنامجا لطباعة تقريب رقم كسري ، (لا تستخدم وظيفة جاهزة)

المدخلات :

Pleas enter a number ?

10.7

المخرجات :

My round Result : 11

C++ round Result : 11

ماذا يعمل ؟	Function & Procedures ...	م
الحصول على الكسور فقط	<code>float</code> GetFractionPart	١
تقريب	<code>int</code> MyRound	٢
قراءة رقم	<code>float</code> ReadNumber	٣

```

#include <iostream>
#include <cmath>

using namespace std;

float GetFractionPart(float Number)
{
    return Number - int(Number) ;
}

int MyRound(float Number)
{
    int IntPart;
    IntPart = int(Number);

    float FractionPart = GetFractionPart(Number);

    if (abs(FractionPart) >= .5)
    {
        if (Number > 0)
            return ++IntPart;
        else
            return --IntPart;
    }
    else
    {
        return IntPart;
    }
}

float ReadNumber()
{
    float Number;
    cout << "Pleas enter a number ? " << endl;
    cin >> Number;

    return Number;
}

int main()
{
    float Number = ReadNumber();

    cout << "My round Result : " << MyRound(Number) << endl;
    cout << "C++ round Result : " << round(Number) << endl;

}

```

حل المشكلة ٤٨

اكتب برنامجا لطباعة تقريب رقم كسري لأدنى رقم صحيح ، (لا تستخدم وظيفة جاهزة)

المدخلات :

Pleas enter a number ?

10.7

المخرجات :

My floor Result : 10

C++ floor Result : 10

ماذا يعمل ؟	Function & Procedures ...	م
تقريب لأدنى	<code>int</code> MyFloor	١
قراءة رقم	<code>float</code> ReadNumber	٢

```
#include <iostream>
#include <cmath>

using namespace std;

int MyFloor(float Number)
{
    if (Number > 0)
        return int(Number);
    else
        return int(Number) -1;
}

float ReadNumber()
{
    float Number;
    cout << "Pleas enter a number ? " << endl;
    cin >> Number;

    return Number;
}

int main()
{
    float Number = ReadNumber();

    cout << "My floor Result : " << MyFloor(Number) << endl;
    cout << "C++ floor Result : " << floor(Number) << endl;
}
```


حل المشكلة ٤٩

اكتب برنامجا لطباعة تقريب رقم كسري لأعلى رقم صحيح ، (لا تستخدم وظيفة جاهزة)

المدخلات :

Pleas enter a number ?

-10.3

المخرجات :

My Ceil Result : -10

C++ Ceil Result : -10

ماذا يعمل ؟	Function & Procedures ...	م
الحصول على الكسور فقط	<code>float</code> GetFractionPart	١
تقريب لأعلى	<code>int</code> MyCeil	٢
قراءة رقم	<code>float</code> ReadNumber	٣

```
#include <iostream>
#include <cmath>
using namespace std;

float GetFractionPart(float Number)
{
    return Number - int(Number);
}

int MyCeil(float Number)
{
    if (abs(GetFractionPart(Number)) > 0)
    {
        if (Number > 0)
            return int(Number) + 1;
        else
            return int(Number);
    }
    else
    {
        return Number;
    }
}

float ReadNumber()
{
    float Number;
    cout << "Pleas enter a number ? " << endl;
    cin >> Number;

    return Number;
}

int main()
{
    float Number = ReadNumber();

    cout << "My Ceiling Result : " << MyCeil(Number) << endl;
    cout << "C++ Ceiling Result : " << ceil(Number) << endl;
}
```

حل المشكلة ٥٠

اكتب برنامجاً لطباعة الجذر التربيعي ، (لا تستخدم وظيفة جاهزة)

المدخلات :

Pleas enter a number ?

25

المخرجات :

My sqrt Result : 5

C++ sqrt Result : 5

ماذا يعمل ؟	Function & Procedures ...	م
الجذر التربيعي	<code>int</code> MySqrt	١
قراءة رقم	<code>float</code> ReadNumber	٢

```
#include <iostream>
#include <cmath>

using namespace std;

int MySqrt(float Number)
{
    return pow(Number , 0.5);
}

float ReadNumber()
{
    float Number;
    cout << "Pleas enter a number ? " << endl;
    cin >> Number;

    return Number;
}

int main()
{
    float Number = ReadNumber();

    cout << "My Sqrt Result : " << MySqrt(Number) << endl;
    cout << "C++ Sqrt Result : " << sqrt(Number) << endl;

}
```