

النوت 20 صفحة كمبيوتر
سحب ملونة دينار كويتي واحد
سحب أبيض وأسود ربع دينار فقط

C++ Programming

Week 6

One Dimensional Arrays

(الشرح والأفكار الرئيسية)

نوت السبب ++

يحتوي على شرح المواضيع
وأمثلة للأفكار الرئيسية من
هواميركات وامتحانات سابقة.

نوت السبب ++

تتكون النوت من عشرة أسابيع.
نوت كل أسبوع يحتوي على الشرح
وتمارين من امتحانات سابقة.

لماذا لا تقتنى الأحدث؟

النوت يتم تنقيحها وتجديدها نهاية
كل أسبوع، راجع eng-hs.net
للتأكد من شرائك الإصدار الأحدث.

(لمن يريد المزيد)

يتوفر على الموقع ملفات الأسابيع
لتمارين وأمثلة إضافية من واقع
امتحانات سابقة.

لن يمكنك تحقيق شيء
تعتقد أنك عاجز عن تحقيقه.

لاستلام نسخ إلكترونية من نوات الموقع مجاناً (شرح وتمارين محلولة) أو (تمارين وأمثلة إضافية) أو (امتحانات سابقة) على إيميلك قم بزيارة eng-hs.net

لطلب نوات الموقع مطبوعة ملونة مجاناً من تصوير الفرع أمام هندسة أسفل صالون رنيم أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو قم بزيارة eng-hs.net

النوات متوفرة مجاناً بالموقعين eng-hs.net, eng-hs.com

(1)

م. حمادة شعبان 260 4444 9 info@eng-hs.com

Best-students (try 1)

Write a program that reads integer marks for 10 students, then print how many of these marks are greater than or equal to their average.

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

Enter 10 marks: **8 5 4 7 9 10 4 5 6 3**
 Average: **6.1**
 Best students:

```
int main ()
{
    int x, i, sum = 0, count = 0;

    double avg;

    cout << "Enter 10 marks: ";

    for (i = 1; i <= 10; i++)
    {
        cin >> x;

        sum += x;
    }

    avg = sum / 10.0;

    cout << "Average: " << avg << endl;

    if (x >= avg)
        count ++;

    cout << "Best students: " << count << endl;

    return 0;
}
```

هذا الحل خطأ لأنه يسأل عن
 آخر قيمة للـ x مرة واحدة
 والناتج سيكون (0) لأن
 x = 3 وهي أقل من avg

لن يموت أحد بدلاً منك فلتحيي
 حياتك بالطريقة التي تراها أنت
 مناسبة لا كما يراها هم.

Best-students (try 2)

Write a program that reads integer marks for 10 students, then print how many of these marks are greater than or equal to their average.

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

```
int main ()
```

```
{
```

```
    int x, i, sum = 0, count = 0;
```

```
    double avg;
```

```
    cout << "Enter 10 marks: ";
```

```
    for (i = 1; i <= 10; i++)
```

```
    {
```

```
        cin >> x;
```

```
        sum += x;
```

```
    }
```

```
    avg = sum / 10.0;
```

```
    cout << "Average: " << avg << endl;
```

```
    for (i = 1; i <= 10; i++)
```

```
        if (x >= avg)
```

```
            count++;
```

```
    cout << "Best students: " << count << endl;
```

```
    return 0;
```

```
}
```

Enter 10 marks: 8 5 4 7 9 10 4 5 6 3
Average: 6.1
Best students:

هذا الحل خطأ لأنه يسأل عن
آخر قيمة لـ x عشر مرات
والناتج سيكون (0) لأن
x = 3 وهي أقل من avg

من العجيب أن كل مشكلة يصاحبها
فرصة تساويها أو تفوق عليها.

Best-students (try 3)

Write a program that reads integer marks for 10 students, then print how many of these marks are greater than or equal to their average.

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

Enter 10 marks: **8 5 4 7 9 10 4 5 6 3**
Average: **6.1**
Best students:

```
int main ()
{
```

```
    int x, i, sum = 0, count = 0;
```

```
    double avg;
```

```
    cout << "Enter 10 marks: ";
```

```
    for (i = 1; i <= 10; i++)
```

```
    {
```

```
        cin >> x;
```

```
        sum += x;
```

```
        if (x >= avg)
```

```
            count ++;
```

```
    }
```

```
    avg = sum / 10.0;
```

```
    cout << "Average: " << avg << endl;
```

```
    cout << "Best students: " << count << endl;
```

```
    return 0;
```

```
}
```

هذا الحل خطأ لأن المقارنة مع
قيمة avg الغير محسوبة
والناتج سيكون (10) لأن كل
قيم الـ x ستكون أكبر من avg

من أكبر متع الحياة القيام بما
يصفه الناس أنه مستحيل.

Best-students (try 4)

Write a program that reads integer marks for 10 students, then print how many of these marks are greater than or equal to their average.

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

```
int main ()
{
```

```
    int x, i, sum = 0, count = 0;
```

```
    double avg;
```

```
    cout << "Enter 10 marks: ";
```

```
    for (i = 1; i <= 10; i++)
```

```
    {
```

```
        cin >> x;
```

```
        sum += x;
```

```
        avg = sum / 10.0;
```

```
        if (x >= avg)
```

```
            count ++;
```

```
    }
```

```
    cout << "Average: " << avg << endl;
```

```
    cout << "Best students: " << count << endl;
```

```
    return 0;
```

```
}
```

Enter 10 marks: 8 5 4 7 9 10 4 5 6 3
Average: 6.1
Best students:

هذا الحل خطأ لأنه يقارن
مع قيم avg غير صحيحة
والناتج سيكون (7) حسب
قيم x الأكبر من avg

i	x	sum	avg	count
1	8	8	0.8	1
2	5	13	1.3	2
3	4	17	1.7	3
4	7	24	2.4	4
5	9	33	3.3	5
6	10	43	4.3	6
7	4	47	4.7	
8	5	52	5.2	
9	6	58	5.8	7
10	3	61	6.1	

نصف من تعرفهم لا يهتمون بشكواك، النصف الآخر
يسعدهم وقوعك بمشكلة، فما جدوى الشكوى؟

Best-students (*correct-answer*)

Write a program that reads integer marks for 10 students, then print how many of these marks are greater than or equal to their average.

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

```
int main ()
{
```

Enter 10 marks: **8** **5** **4** **7** **9** **10** **4** **5** **6** **3**
 Average: **6.1**
 Best students: **4**

```
int x [10], i, sum = 0, count = 0;
double avg;
```

x

8	5	4	7	9	10	4	5	6	3
0	1	2	3	4	5	6	7	8	9

```
cout << "Enter 10 marks: ";
```

```
for (i = 0; i < 10; i++)
```

```
{
    cin >> x [i];

    sum += x [i];
}
```

إذا لدينا مصفوفة عشرة حدود
فإن فهرس أماكنها من (0) إلى
(9) وليس من (1) إلى (10).

```
avg = sum / 10.0;
```

```
cout << "Average: " << avg << endl;
```

```
for (i = 0; i < 10; i++)
```

```
if (x [i] >= avg)
    count ++;
```

```
cout << "Best students: " << count << endl;
```

```
return 0;
```

```
}
```

الكثير يفقدون صحتهم أثناء
كسبهم المال ثم ينفقون هذا
المال محاولين استعادة صحتهم.

1-D Arrays (*Print-Reverse*)

Write a program that reads 10 integers and prints them in a reverse order.

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

Enter 10 integers: 2 1 5 -7 8 10 3 4 6 17
Printing reverse: 17 6 4 3 10 8 -7 5 1 2

```
int main ( )
{
    int x [10], i;

    cout << "Enter 10 integers: " ;

    for (i = 0; i < 10; i ++)
        cin >> x [i];

    cout << "Printing reverse: ";

    for (i = 9; i >= 0; i --)
        cout << x [i] << "    ";

    cout << endl;

    return 0;
}
```

النوم جيد، والموت أفضل منه،
لكن بالطبع أفضل شيء ألا تكون
قد ولدت من الأساس.

1-D Arrays with functions (*Sum-Evens*)

Write a function that gets an integer array and its number of elements. The function should return the sum of the even numbers in the array.

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

Enter 10 integers: **2 1 5 7 8 10 3 4 9 6**
Sum of Even Numbers = **30**

```
int sum_evens (int [ ], int);
```

```
int main ( )
```

```
{
```

```
    int x [10], i, sum;
```

```
    cout << "Enter 10 integers: ";
```

```
    for (i = 0; i <= 9; i ++)
```

```
        cin >> x [i];
```

```
    sum = sum_evens (x, 10);
```

```
    cout << "Sum of Even Numbers = " << sum << endl;
```

```
    return 0;
```

```
}
```

بما أن الدالة تستقبل عدد الحدود خلال معامل (n) فإنها تصلح لأي عدد من الحدود.

```
int sum_evens (int x [ ], int n)
```

```
{
```

```
    int sum = 0;
```

```
    for (int i = 0; i < n; i ++)
```

```
        if (x [i] % 2 == 0)
```

```
            sum += x [i];
```

```
    return sum;
```

```
}
```

الواقع يقول: الفرح لا يدوم، والحزن كذلك، أقصد الحياة كلها لا تدوم.

1-D Arrays with functions (*Average-Odds*)

Write a function that gets an integer array and its number of elements. The function should return the average of the odd numbers in the array.

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

Enter 10 integers: **2 3 5 7 8 10 3 4 9 6**
Average of odd numbers = **5.40**

```
double avg_odds (int [ ], int);
```

```
int main ( )
```

```
{
    int x [10], i;

    cout << "Enter 10 integers: ";

    for (i = 0; i < 10; i ++ )
        cin >> x [i];

    cout << "Average of odd numbers = " << avg_odds (x, 10) << endl;

    return 0;
}
```

```
double avg_odds (int x [ ], int n)
```

```
{
    int sum = 0, count = 0;

    for (int i = 0; i < n; i ++ )
        if (x [i] % 2 != 0)
        {
            sum += x [i];

            ++ count;
        }

    if (count == 0)
        return 0;

    return double (sum) / count;
}
```

قبل إجراء عملية التقسيم يجب التأكد
أن قيمة المقام لا تساوي صفراً.

إذا أقيمت الدنيا على أحد أعارته
محاسن غيره، وإذا أدبرت عنه
سلبته محاسن نفسه.

1-D Arrays with functions (*Average*)

Write a function that gets an integer array and its number of elements. The function should return the average of the array elements.

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

Enter 10 marks: **3 5 4 7 9 10 4 5 6 8**
Average = **6.1**

```
double average (int x [ ], int n)
```

```
{
    int i, sum = 0;

    for (i = 0; i < n; i++)
        sum += x [i];

    return sum * 1.0 / n;
}
```

إذا المتوسط يتعلق بكل الحدود (n)
فلا داعي لإنشاء عداد لعدد الحدود.

```
int main ( )
```

```
{
    int x [10], i;

    cout << "Enter 10 integers: ";

    for (i = 0; i < 10; i++)
        cin >> x [i];

    cout << "Average = " << average (x, 10) << endl;

    return 0;
}
```

تذكر أن أماكن المصفوفات تبدأ
دائماً من (0) وليس (1).

إن الحياة ليست بروفة لتجربة
ثوب، إنها أصلية وواحدة فقط.

1-D Arrays with functions (*Search-Key*)

Write a function that gets an array, its number of elements and a *key*. The function should return true if *key* exists in the array, otherwise it returns false.

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

Enter 7 numbers: 2.5 1.7 7.2 15.5 17.8 7.2 1.1
Enter the search key: 17.8
Exists

```
bool Search (double x [ ], int n, double key)
```

```
{
    for (int i = 0; i < n ; i++)
        if (x [i] == key)
            return true;
    return false;
}
```

Enter 7 numbers: 2.5 1.7 7.2 15.5 17.8 7.2 1.1
Enter the search key: 3.6
Does not exist

```
int main ( )
```

```
{
    double x [7], key;
    bool y;

    cout << "Enter 7 numbers: ";
    for (int i = 0; i < 7; i++)
        cin >> x [i];

    cout << "Enter the search key: ";
    cin >> key;

    y = Search (x, 7, key);
    if (y == false)
        cout << "Does not Exist\n";
    else
        cout << "Exists\n";

    return 0;
}
```

عدد عناصر المصفوفة رقم صحيح
بغض النظر عن نوعية عناصرها.

إن العالم ليس مكاناً جميلاً،
وكذلك ليس مكاناً سيئاً، إنه
مكان محايد في أفضل الأحوال.

1-D arrays with functions (*Search-Key-Count*)

Write a function that gets an array, its number of elements and a key.
The function should return how many times the key presents in the array.

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

Enter 7 numbers: *2.5 1.7 7.2 15.5 17.8 7.2 1.1*
Enter the search key: *7.2*
Exists 2 times

```
int search (double [ ], int, double);
```

```
int main ( )
```

```
{
```

```
    double x [7], key;
    int c;
```

Enter 7 numbers: *2.5 1.7 7.2 15.5 17.8 7.2 1.1*
Enter the search key: *3.6*
Does not exist

```
    cout << "Enter 7 numbers: ";
    for (int i = 0; i < 7; i = i+1)
        cin >> x [i];
```

```
    cout << "Enter the search key: ";
    cin >> key;
```

```
    c = search (x, 7, key);
```

```
    if (c > 0)
        cout << "Exists " << c << " times" << endl;
    else
        cout << "Does not exist" << endl;
```

```
    return 0;
```

```
}
```

```
int search (double x [], int n, double key)
```

```
{
```

```
    int count = 0;
    for (int i = 0; i < n; i++)
        if (x [i] == key)
            ++ count;
```

```
    return count;
```

```
}
```

أجمل الأنهار لم نرها بعد،
أجمل الكتب لم نقرأها بعد،
أجمل أيام حياتنا لم تأت بعد.

1-D Arrays with functions (*Minimum*)

Write a function that gets an array and its number of elements. The function should return the minimum element in the array.

Enter 7 numbers: 2.5 1.9 7.2 15.5 1.8 7.2 11.1
The minimum is: 1.8

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

```
double minimum (double x [ ], int n)
```

```
{
    double min = x [0];

    for (int i = 1; i < n; i ++)
        if (x [i] < min)
            min = x [i];

    return min;
}
```

المقارنة تبدأ من (1) لأننا اعتبرنا
أول عنصر x [0] هو (min).

```
int main ( )
{
    double x [7], min;

    cout << "Enter 7 numbers: ";

    for (int i = 0; i < 7; i ++)
        cin >> x [i];

    min = minimum (x, 7);

    cout << "The minimum is: " << min << endl;

    return 0;
}
```

أغلب البشر يحيون في سباق
ليس فيه رابح وليس له نهاية،
ولا يمكنهم الخروج منه.

1-D Arrays with functions (*Minimum index*)

Write a function that gets an array and its number of elements. The function should return the index of minimum element in the array.

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

Enter 7 numbers: 2.5 1.9 7.2 15.5 1.8 7.2 11.1
The index is: 4
The minimum is: 1.8

```
int minimum (double x [ ], int n)
{
    int min = 0;

    for (int i = 1; i < n; i++)
        if (x [i] < x [min])
            min = i;

    return min;
}
```

المقارنة تتم بين القيم، لكن الذي يتم
تخزينه هو مكان العنصر الأقل.

```
int main ( )
{
    double x [7];

    int min;

    cout << "Enter 7 numbers: ";

    for (int i = 0; i < 7; i++)
        cin >> x [i];

    min = minimum (x, 7);

    cout << "The index is: " << min << endl;

    cout << "The minimum is: " << x [min] << endl;

    return 0;
}
```

أغلب الناس تبذل جهداً بالحد
الذي يكفي لبقيائهم، مجرد بقاء.

1-D Arrays with functions (*Equal-Arrays*)

Write a function that gets two arrays of the same size, the function should return true if they are exactly equal, otherwise it will return false.

```
#include <iostream>
using namespace std;
bool equal_arrays (int [ ], int [ ], int);
```

```
int main ( )
{
    int x [100], y[100], n, i;
    bool equal;

    cout << "Enter the number of integers: ";
    cin >> n;

    cout << "Enter the elements of the first array: ";
    for (i = 0; i < n; i++)
        cin >> x [i];

    cout << "Enter the elements of the second array: ";
    for (i = 0; i < n; i++)
        cin >> y [i];

    equal = equal_arrays (x, y, n);

    if (equal == true)
        cout << "Equal\n";
    else
        cout << "Not equal\n";

    return 0;
}
```

يتم مقارنة مصفوفتين فقط إذا
كان عدد العناصر بها متساو.

```
bool equal_arrays (int x [ ], int y [ ], int n)
{
```

```
    for (int i = 0; i < n; i++)
```

```
        if (x [i] != y [i])
```

```
            return false;
```

```
        return true;
}
```

أن نحيا ذلك شيء نادر الحدوث، الحقيقة أن
معظم الناس متواجدون فقط، هذا كل شيء.

1-D Arrays (Output)

What is the output of the following codes?

T	h	i	s		i	s		a	n		e	a	s	y		e	x	a	m	\0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	

```
a.int main( )
{
    char str[] = "This is an easy exam";

    int i = 14;
    while(str[i] != 'x')
    {
        cout << str[i];
        i -= 3;
        cout << str[i];
        i += 2;
        cout << str[i++];
        i += 3;
    }
}
```

output

yes

لعكس عناصر مصفوفة يتم عمل swap
لنصفي حدودها الأيمن والأيسر.

```
b.int main( )
{
    int i, hold, n = 7;
    int x[7] = {6, 3, 2, 7, 1, 8, 4};

    for (i = 0 ; i < n/2 ; i++)
    {
        hold = x[i];
        x[i] = x[n-1-i];
        x[n-1-i] = hold;
    }

    cout << "new x is: ";
    for (i = 0; i < n; i++)
        cout << x[i] << " ";
    cout << endl;
    return 0;
}
```

6	3	2	7	1	8	4
4	8	1		2	3	6

i	hold
0	6
1	3
2	2

output

new x is: 4 8 1 7 2 3 6

تتسم الحياة دائماً بالغموض والمشاكل فإذا رأيت الطريق
أمامك ممهداً فاعلم أنه ليس الطريق الصحيح.

Use a single-subscripted array to solve the following problem. A company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9 percent of their gross sales for that week. For example, a salesperson who grosses \$5000 in sales in a week receives \$200 plus 9 percent of \$5000, or a total of \$650. Write a program (using an array of counters) that determines how many of the sales people earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):

- a) \$200-\$299
- b) \$300-\$399
- c) \$400-\$499
- d) \$500-\$599
- e) \$600-\$699
- f) \$700-\$799
- g) \$800-\$899
- h) \$900-\$999
- i) \$1000 and over.

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

```
int main ( )
```

```
{
```

```
    int A[11] = {0};
    int salary, gross;
```

```
    cout << "Enter employee's gross or 0 to end: ";
    cin >> gross;
```

```
    while ( gross > 0 )
```

```
    {
```

```
        salary = 200 + 0.09 * gross;
```

```
        A[salary / 100]++;
```

```
        cout << "Enter employee's gross or 0 to end: ";
```

```
        cin >> gross;
```

```
    }
```

في النهاية، العبرة ليست بالسنوات التي عشتها، بل في نبض الحياة في تلك السنوات.

```

cout << "\nSalary range" << setw(15) << "Employees\n";

for ( int i = 2; i < 10; i++ )
    cout << "$" << i * 100 << "-$" << i * 100 + 99 << "    " << setw(8)
        << A[i] << endl;

cout << "$1000 and over    " << A[10] << endl;

return 0;
}

```

Enter employee's gross or 0 to end: 5000
 Enter employee's gross or 0 to end: 4360
 Enter employee's gross or 0 to end: 250
 Enter employee's gross or 0 to end: 8346
 Enter employee's gross or 0 to end: 340
 Enter employee's gross or 0 to end: 1090
 Enter employee's gross or 0 to end: 6356
 Enter employee's gross or 0 to end: 34263
 Enter employee's gross or 0 to end: 870
 Enter employee's gross or 0 to end: 500
 Enter employee's gross or 0 to end: 2300
 Enter employee's gross or 0 to end: 9400
 Enter employee's gross or 0 to end: 436
 Enter employee's gross or 0 to end: 642
 Enter employee's gross or 0 to end: 0

Salary range	Employees
\$200-\$299	7
\$300-\$399	0
\$400-\$499	1
\$500-\$599	1
\$600-\$699	1
\$700-\$799	1
\$800-\$899	0
\$900-\$999	1
\$1000 and over	1

كل الوقائع تشير إلى أن أغلب الناس موجودون في هذا العالم لكي لا تفعل شيئاً.

Use a single-subscripted array to solve the following problem. Read in 20 numbers, each of which is between 10 and 100, inclusive. As each number is read, print it only if it is not a duplicate of a number already read. Provide for the “worst case” in which all 20 numbers are different. Use the smallest possible array to solve this problem.

```
# include <iostream>
using namespace std;
bool search ( int [], int, int );
int main ( )
{
    int x[20];
    int num, n = 0;;
    bool found;p;
    for ( int i = 0; i < 20; i++ )
    {
        cout << "Enter a number between 10-100: ";
        cin >> num;
        if ( num >= 10 && num <= 100 )
        {
            found = search ( x, n, num );
            if ( found == false )
            {
                x[n] = num;  n++;
                cout << "The number " << num << " is accepted\n";
            }
        }
        else
        {cout << "Invalid entry, re-enter again.\n";  i--;}
    }
    return 0;
}

bool search ( int x[ ], int size, int key )
{
    for ( int i = 0; i < size; i++ )
        if ( x[i] == key )
            return true;
    return false;
}
```

للحياة معان كثيرة لم يتعرف
أغلب الناس على معظمها.

Write a program that simulates the rolling of two dice. Your program should roll the two dice 36,000 times. Use a single-subscripted array to show how many times each possible sum appears. Print the results in a tabular format.

```
#include <iostream>
#include <cstdlib>      // for srand ( )
#include <ctime>         // for time ( 0 )
#include <iomanip>        // for setw ( )
using namespace std;

int main ( )
{
    int sum;
    int a[13] = {0};

    srand ( time ( 0 ) );

    for ( int i = 1; i <= 36000; i++ )
    {
        sum = 2 + rand ( ) % 6 + rand ( ) % 6;
        ++a[sum];
    }

    cout << "Sum" << setw(9) << "Times\n";

    for ( int i = 2; i <= 12; i++ )
        cout << setw(3) << i << setw(10) << a[i] << endl;

    return 0;
}
```

Sum	Times
2	99
3	183
4	303
5	385
6	517
7	565
8	504
9	423
10	321
11	194
12	106

يدرك الإنسان أهمية الوقت عندما
يعيش اللحظات الأخيرة في حياته،
أغلبنا ينتظر هذا الوقت كي يندم.