

BÀI THỰC HÀNH SỐ 10 :: CẤU TRÚC MẢNG (ARRAY)

MỤC TIÊU

1. Định nghĩa kiểu mảng, khai báo biến mảng và truy cập trực tiếp đến phần tử mảng hoặc duyệt tuần tự trên mảng.
2. Tạo dữ liệu mảng bằng cách gán trực tiếp, nhận từ bàn phím hoặc sinh giá trị ngẫu nhiên thỏa mãn một số yêu cầu cho trước.
3. Các thao tác duyệt mảng một chiều: viếng thăm, tìm kiếm, chèn và xóa phần tử.

NỘI DUNG THỰC HÀNH

Vấn đề 1: Nhập và Xuất dữ liệu mảng

1. Viết chương trình nhập dữ liệu cho mảng A chứa tối đa n số nguyên, sau đó in ra màn hình các phần tử của mảng với k số trên một dòng (các phần tử cách nhau 1 tab, xem hình).
2. Viết chương trình sinh ngẫu nhiên n số nguyên và lưu vào mảng B, hiển thị lên màn hình các số chứa trong B, các phần tử cách nhau 1 ký tự trắng.
3. Viết chương trình nhập vào dãy n số nguyên C, các phần tử của C có giá trị nằm trong đoạn [a, b] với a, b nhập vào từ bàn phím ($a \leq C[i] \leq b$, với mọi $i = 0, \dots, n-1$). In ra màn hình các phần tử của dãy C, theo thứ tự ngược lại với quá trình nhập.

```

D:\REC Document\KyThuatLapTrinh\A...
So luong phan tu cua mang n=8
a[0]=1
a[1]=2
a[2]=3
a[3]=4
a[4]=5
a[5]=6
a[6]=7
a[7]=8
So phan tu can in tren 1 dong?3
CAC PHAN TU CUA DAY:
      2      3
1      5      6
4      8      -
  
```

Vấn đề 2: Duyệt mảng (viết thăm tất cả các phần tử của mảng)

1. Định nghĩa hàm **int songuyeto(int k)** trả về 1 nếu k là số nguyên tố và trả về 0 nếu k không phải số nguyên tố.
2. Định nghĩa hàm **int sochinhphuong(int k)** trả về 1 nếu k là số chính phương và trả về 0 nếu k không phải là số chính phương. (k là số chính phương nếu tồn tại h để $h^2 = k$)
3. Định nghĩa hàm **int sodoixung(int k)** trả về 1 nếu k là số đối xứng (đảo thứ tự các số của k từ phải sang trái thì được chính số k, chẳng hạn 12321 là số đối xứng) và trả về 0 nếu k không đối xứng.
4. Định nghĩa hàm **int soboi3(int k)** trả về 1 nếu tổng các chữ số của k chia hết cho 3 (ví dụ 123 có $1+2+3=6$, 6 chia hết cho 3 nên 123 là số bội 3)
5. Viết chương trình **duyetmang.cpp** thực các yêu cầu sau:
 - Tạo dãy số A chứa n phần tử là các số tự nhiên không vượt quá n
 - In tất cả các số nguyên tố xuất hiện trong dãy A, 10 số trên 1 dòng
 - Tổng của các số chính phương có trong dãy A
 - Số lượng (đếm) các số đối xứng có trong dãy A

- Trung bình cộng các số bội 3 của dãy A (nếu không có số bội 3 thì trung bình cộng là 0)

Yêu cầu:

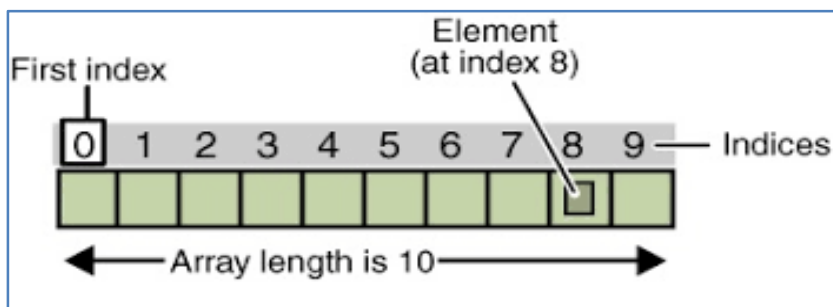
- Vẽ sơ đồ khối cho mỗi yêu cầu
- Cài đặt chương trình, sử dụng các hàm đã định nghĩa
- Lập test_case cho bài toán trên.

Vấn đề 3: Thao tác trên mảng một chiều

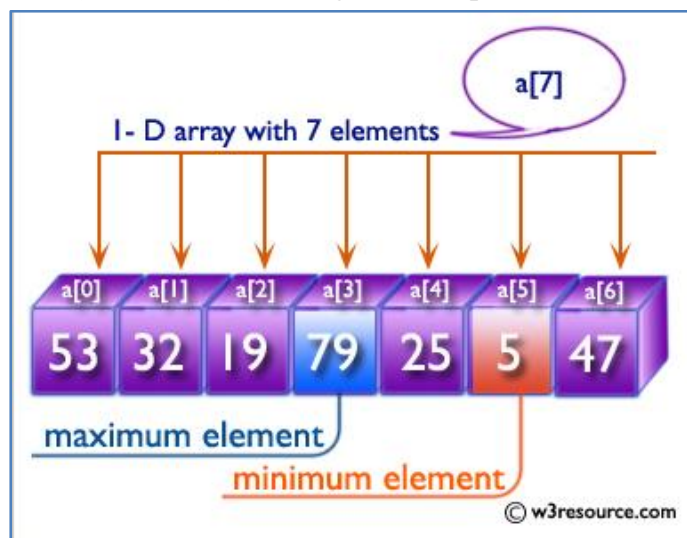
Viết chương trình *mang1chieu.cpp* thực hiện các công việc sau trên mảng 1 chiều:

- Xây dựng hàm ***void input(int A, int &n)*** tạo mảng A chứa n phần tử là các số nguyên
- Xây dựng hàm ***void output(int A, int n)*** hiển thị các phần tử của dãy A, giá trị các phần tử viết cách nhau 1 ký tự trắng.

PRIME NUMBERS									
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100



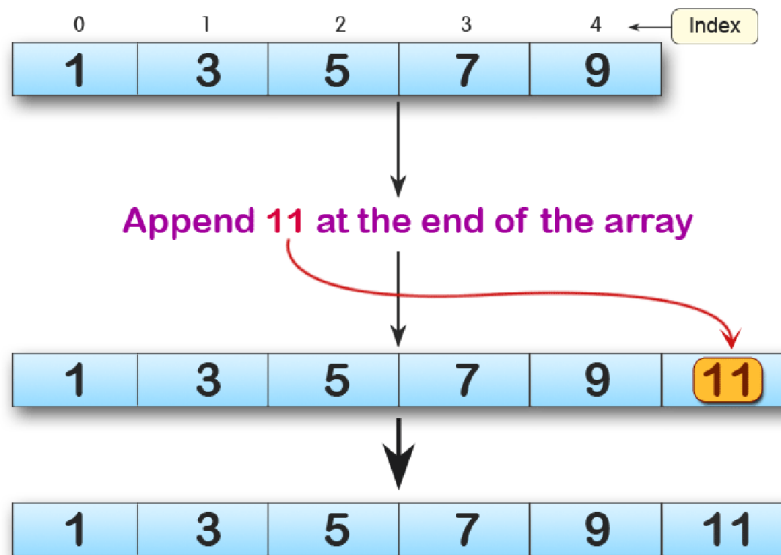
- Xây dựng hàm ***int max(int A, int n)*** trả về giá trị của phần tử lớn nhất của mảng A
- Xây dựng hàm ***int min(int A, int n)*** trả về giá trị của phần tử nhỏ nhất của mảng A



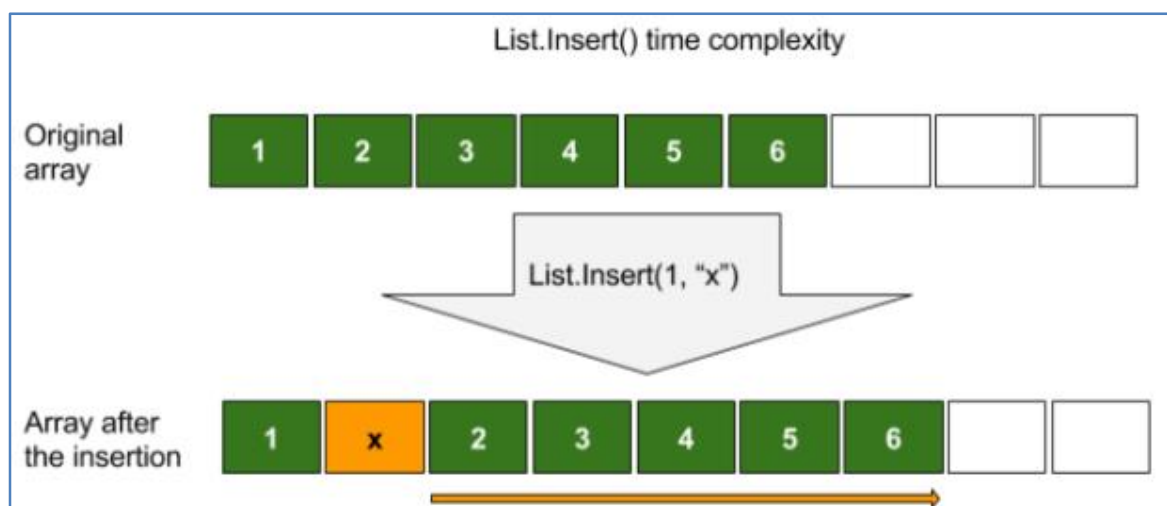
- Xây dựng hàm ***int search(int A, int n, int x)*** trả về vị trí xuất hiện giá trị x trong dãy A (trả về i nếu $A[i]=x$) hoặc trả về -1 nếu x không xuất hiện trong dãy (trường hợp x xuất hiện nhiều lần trong dãy thì trả về vị trí xuất hiện đầu tiên).
(Trong minh họa dưới đây, dãy A có 12 phần tử và $x=62$ tìm thấy ở vị trí số 4)



- Xây dựng hàm ***void append(int A, int &n, int x)*** để bổ sung x vào cuối dãy A (x được thêm vào cuối dãy A. (Xem mô tả bên dưới, giá trị 11 được bổ sung vào cuối dãy A)

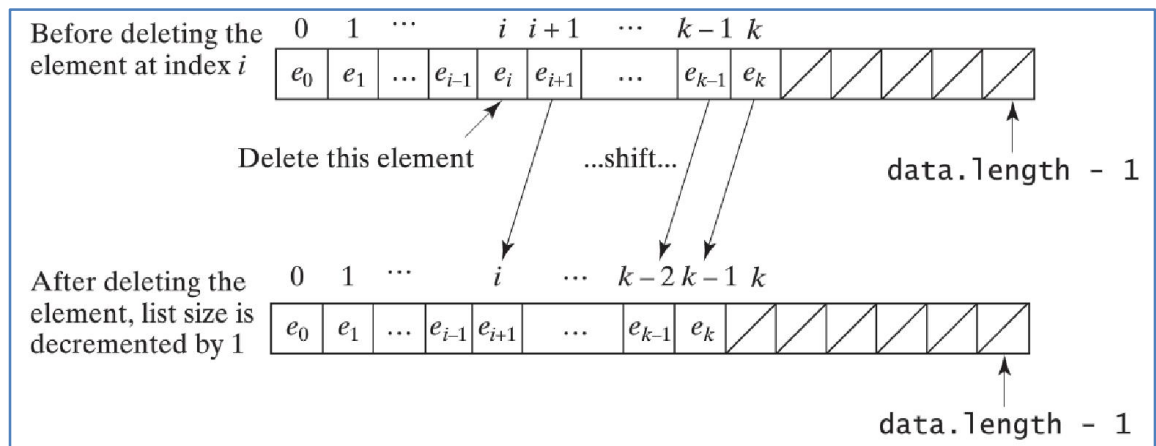


- Xây dựng hàm ***void insert(int A, int &n, int k, int x)*** để chèn x vào vị trí thứ k trong dãy A (x được đưa vào để trở thành phần tử thứ k của dãy A)



Trong ví dụ ở trên, x được chèn vào vị trí 1, khi đó từ phần tử thứ 1 trở đi phải dịch qua phải 1 vị trí để chèn x vào. Trên thực tế, chúng ta cần xuất phát từ cuối dãy đến vị trí thứ k và dịch chuyển phải 1 phần tử. Sau khi chèn, số phần tử của dãy tăng lên 1.

- Xây dựng hàm ***void delete(int A, int &n, int k)*** để xóa phần tử thứ k của dãy A. Trong minh họa dưới đây, phần tử e_k bị xóa và khi đó số phần tử của dãy giảm đi 1 (nhưng kích thước ô nhớ - độ dài của mảng không đổi)



- Xây dựng hàm ***void sort(int A, int n)*** sắp xếp mảng A để được dãy số có thứ tự tăng.

	3	10	4	6	2	6	15	3	9	7
Bước 1	2	3	10	4	6	3	6	15	7	9
Bước 2		3	3	10	4	6	6	7	15	9
Bước 3			3	4	10	6	6	7	9	15
Bước 4				4	6	10	6	7	9	15
Bước 5					6	6	10	7	9	15
Bước 6						6	7	10	9	15
Bước 7							7	9	10	15
Bước 8								9	10	15
Bước 9									10	15
Kết quả	2	3	3	4	6	6	7	9	10	15

Yêu cầu 1: Cài đặt chương trình có menu hiển thị như bên dưới:

- Tạo dãy số
- Hiển thị dãy số
- Tìm Max
- Tìm Min
- Tìm phần tử X
- Bổ sung phần tử
- Chèn phần tử vào vị trí k
- Xóa phần tử thứ k
- Sắp xếp dãy số
- Thoát

Chọn chức năng bằng cách nhập 1 số từ 1 đến 10:

Khi người dùng nhập số từ 1 đến 10 thì thực hiện chức năng tương ứng, sau khi thực hiện xong thì quay lại Menu (hiển thị menu và cho phép chọn để thực hiện công việc), chương trình chỉ kết thúc nếu người dùng nhập số 10.

Lưu ý:

- Khi người dùng chọn 5 thì yêu cầu nhập giá trị X và hiển thị thông báo tìm thấy X ở vị trí nào hoặc thông báo không tìm thấy
- Khi người dùng chọn 6 thì yêu cầu nhập giá trị X để thêm X vào cuối dãy.
- Khi người dùng chọn 7 thì yêu cầu nhập X và k để chèn X vào vị trí k trong dãy
- Khi người dùng chọn 8 thì yêu cầu nhập k để xác định vị trí phần tử cần xóa

Yêu cầu 2: Vẽ sơ đồ khối của mỗi thao tác:

- Tìm min/max
- Tìm x trong dãy A
- Bổ sung x vào cuối dãy A
- Xóa phần tử ở vị trí k trong A
- Chèn giá trị x vào A ở vị trí k
- Sắp xếp để A có trật tự (tăng dần, hoặc giảm dần)

BÀI TẬP

Bài 0061

Viết chương trình khởi tạo mảng một chiều gồm 1000 số ngẫu nhiên. Hiển thị giá trị trung bình và giá trị các phần tử mảng, và sau mỗi 10 giá trị hiển thị thì tạm dừng màn hình.

Ghi chú:

```
#include <stdlib.h>
int rand( void );
```

The function rand() returns a pseudorandom integer between zero and **RAND_MAX**.

Bài 0062

Viết chương trình khởi tạo mảng một chiều gồm 10 phần tử, mỗi phần tử có giá trị bằng chỉ số của nó. In ra màn hình dãy số đó.

Bài 0063

Sửa lại chương trình ở Bài 0062, sau khi hiển thị các giá trị khởi tạo, chương trình sẽ sao chép các giá trị vào một mảng mới và thêm vào 10 cho mỗi giá trị. Hiển thị ra màn hình giá trị các phần tử của mảng mới.

Bài 0064

Viết chương trình khởi tạo mảng ngẫu nhiên các số nguyên có kích thước không quá 100 và chúng được sắp xếp theo thứ tự tăng ngay trong quá trình tạo mảng.

Bài 0065

Tạo mảng A gồm n phần tử nguyên được sắp xếp (tăng dần hoặc giảm dần). Nhập vào một số nguyên x, tiến hành chèn số x này vào mảng A sao cho thứ tự trong mảng không thay đổi.

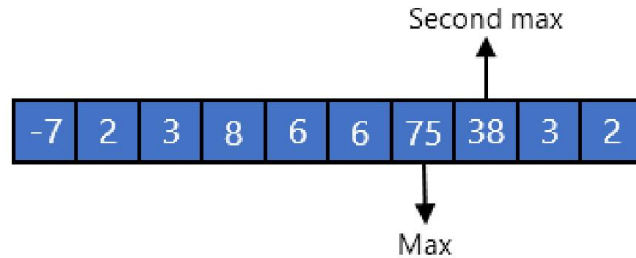
Bài 0066

Tạo mảng A chứa n phần tử là các số nguyên. Nhập vào một số nguyên x và hãy xóa tất cả các phần tử của A có giá trị bằng x.

Lưu ý: In dãy trước và sau khi xóa.

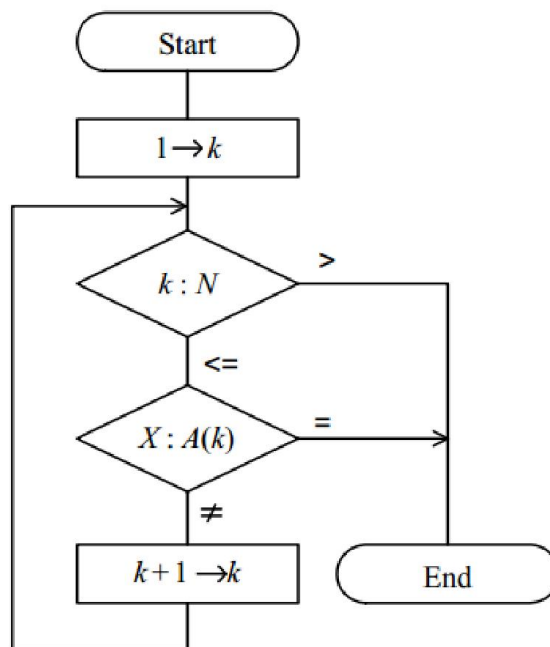
Bài 0067

Tạo mảng A chứa n phần tử là các số nguyên. Tìm số lớn nhất và số lớn thứ 2 của dãy A.



English

Q1. Integers are stored in the 1st to N-th elements of an array A ($N > 1$). The flowchart below shows the process to check which element of the array contains the same value as X. Which of the following correctly describes the execution result of this process?



- a) If the same value as X exists in two places of the array, the 1st and N-th elements, k is set to 1.
- b) If the same value as X exists in two places of the array, the 1st and N-th elements, k is set to N.
- c) If the same value as X is not in the array, k is set to 1.
- d) If the same value as X is not in the array, k is set to N.

Q2. For a two-dimensional integer array a, whose element $a(i,j)$ is of value $2i + j$, what is the value $a(a(1,1) \times 2, a(2,2) + 1)$?

- a) 12
- b) 13
- c) 18
- d) 19