Questions

1. How can we declare an array?
2. What is equal to arr[5]?
3. Are the numbers in the array represented consecutively?
4. Can I change an array's size and why?
5. What is the difference between static and dynamic array?
6. Is array a data type?
7. What is the fastest way to access the middle element in an array (arr) of odd size n ?
8. If you have a circular array (arr) what is the next element of the last one?
9. What is the memory size of int arr[5]?
10. What is the memory size of int arr[5][6]?

11. How can I represent an 5D integer array?

12. What is the name of "[]"?

13. What are the differences between Static and Dynamic arrays ?

14. What is Complexity?

15. Why do we use Big-O notation?

16. The same algorithm can be written with different complexities, give an example?

17. What is Binary Search? How does it work?

18. What is the complexity of a binary search algorithm?

19. To apply Binary Search, your array must be……

20. What are the time and space complexity of the selection sort?

21. What is the best, worst, and average-case time complexity of bubble sort?

22. What is the output ?

ICPC Assuit Community

Newcomers (*Arrays*)

A.

```
#include<iostream>
using namespace std;
int main() {
    int arr[5] = { 0 };
    for (int i = 0; i <= 5; i++)
         cout << arr[i] << " ";
    return 0;
}
```

B.

```
#include <iostream>
using namespace std;
int main() {
    int a[][] = { { 1, 2 }, { 3, 4 } };
    for (int i = 0; i < 2; i++)
         for (int j = 0; j < 2; j++)
              cout << a[i][j] << " ";
    return 0;
}
```

I. 1 2 3 4
II. Compilation Error
III. 4 garbage values
IV. 4 3 2 1

C.

```cpp
#include<iostream>
using namespace std;
int main() {
    int arr[5];
    for (int i = 0; i < 5; i++) {
        arr[i] = i;
    }
    cout << arr;
}
```

D.

```cpp
#include<iostream>
using namespace std;
int main() {
    int n = 5;
    int arr[5];
    for (int i = 0; i <= 5; i++) {
        arr[i] = i + 1;
    }
    arr[--n] / 2;
}
```

E.

```
#include<iostream>
using namespace std;
int main() {
      int arr[5];
      for (int i = 0; i < 5; i++) {
            arr[i] = (3 * i) % 5;
}
      cout << arr[arr[1]];
}
```

F.

```
#include<iostream>
using namespace std;
int arr[100000];
int main() {
    int n;
       n = 5;
    for (int i = 1; i < n; i++) {
         arr[i] = arr[i - 1] + i;
    }
    cout << arr[n - 1];
}
```

G.

```cpp
#include<iostream>
using namespace std;
int main() {
  int arr[1000];
  for (int i = 1; i < 7; i++) {
       arr[i] = arr[i - 1] + i;
  }
  cout << arr[5];
}
```

H.

```cpp
#include<iostream>
using namespace std;
int array1[] = { 1200, 200, 2300, 1230, 1543 };
int array2[] = { 12, 14, 16, 18, 20 };
int temp, result = 0;
int main() {
  for (temp = 0; temp < 5; temp++) {
       result += array1[temp];
  }
  for (temp = 0; temp < 4; temp++) {
       result += array2[temp];     }
  cout << result;
  return 0;}
```

a) 6553
b) 6533
c)6522
d)12200

I.

```
#include<iostream>
using namespace std;
int main() {
    int list[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int i;
    for (i = 0; i < 5; i++) {
         int temp = list[i];
         list[i] = list[9 - i];
         list[9 - i] = temp;
    }
    for (i = 0; i < 10; i++)
         cout << list[i] << "\t";
}
```

J.

```
#include<iostream>
using namespace std;
int n = 5, a[] = { 1, 2, 3, 4, 5 }, fre[10];
int main()
{
    for (int i = 0; i < n; i++)
    {
         int x = a[i];
         x = x % 2;
         if (x)
              x++;
         else
              x += 2;
         fre[x]++;
    }
    cout << fre[2] << "\n";
}
```

K.

```cpp
#include<iostream>
#include<cmath>
using namespace std;
int main() {

    long long n, sum = 0;
    long long nums[8] = { 2,5,6,3,0,8,9,6 };
    for (int i = 0; i < 8; i++) {
         if ((i && (i - 1) && nums[i] - 1) || sum++) {

              sum += nums[i];
              cout << nums[i] << " ";
         }
    }
    cout << sum;
    return 0;
}
```

L.

```cpp
#include <iostream>
using namespace std;
int main()
{
     char str[5] = "ABC";
     cout << str[3];
     cout << str;
     return 0;
}
```

M.

```
#include <iostream>
using namespace std;
int main()
{
    int array[] = { 10,20,30 };
    cout << -2[array];

    return 0;
}
```

a) -15
b) -30
c) Compile time error
d) Garbage error

N.

```
#include <iostream>
using namespace std;
int main()
{
    long num[10] = { 2,4,6,7,8,9,3,4,7,2 };
    for (int i = 0; i < 5; i++);
    for (int l = 0; l < 9; l++)
        if (num[l] > num[l + 1])swap(num[l], num[l + 1]);

    for (int i = 0; i < 10; i++);
        cout << num[i] << " ";

    return 0;
}
```

O.

```cpp
#include<iostream>
using namespace std;
int n = 6, a[] = { 1, 2, 3, 4, 5, 6 };
int binarySearch(int s, int e, int k) {
    int md, ix = -1;
    while (s < e) {
         md = (s + e) / 2;
         if (a[md] == k) {
              ix = md;
              break;
         }
         if (a[md] > k)
              e = md - 1;
         else
              s = md + 1;
    }
    return ix;
}
int main()

{
    cout << binarySearch(0, n - 1, 4) << "\n";
}
```

P.

```
#include<iostream>
using namespace std;
int n = 5, a[] = { 1, 2, 3, 4, 5 };
int binarySearch(int s, int e, int k) {
    int md, ix = -1;
    while (s <= e) {
         md = (s + e) / 2;
         if (a[md] == k) {
              ix = md;
              break;
         }
         if (a[md] > k)
              e = md;
         else
              s = md;
    }
    return ix;
}
int main()

{
    cout << binarySearch(0, n - 1, 5) << "\n";
}
```

23. Problems

A. Given an array, print the elements in the even indices.

B. Given an array, print odd numbers then print even numbers.

C. Given an array, print its elements in reversed order.

D. Given an array, check if it's a palindrome or not.

E. Given 2 matrices, your task is to swap the main diagonal of the matrices with each other.(O(N))

1st array	2nd array			1st array	2nd array
1 2 3	2 4 6	result		2 2 3	1 4 6
4 5 6	3 2 1	⇒	4 2 6	3 5 1	
7 8 9	2 4 8		7 8 8	2 4 9	

F. Given an array, you can change elements from it, print the minimum number of elements you need to change ,to make all elements in the array equal. (0 <= numbers <= 1e5)

G. Given an array A of N integers returns the smallest integer that does not occur in A, as N is an integer within the range [1..100,000] and each element of array A is an integer within the range [0..1,000,00].

H. You have an array of chars, determine if you can create the word "Hello" by using the chars you are given or not.

I. You have an array and your task is to calculate the summation of the maximum value in each prefix of the array and the summation of the minimum value in each prefix of the array.

ICPC Assuit Community

Newcomers (*Arrays*)

ICPC Assiut community

J. A Good number is a number that is composed of two numbers multiplied by 2 ,concatenated then multiplied by 2 one more time .

Ex: a=12 , b=34

Step 1: a=12*2=24 , b=34*2=68

Step 2: num=2468

Step 3: the good =2468*2 ⇒ 4936

K. Given an array *A* of consist of *N* numbers (4 <N<1^5) , (-10^9<A[i]<10^9) and your task is to find the second lowest and the second-highest numbers

L. Write a C++ program to count the number of occurrences of a given number in a sorted array of integers (you have to answer several queries) in O(n*q)

M. You have an array, your task is to find the first number that has been repeated in the array.

N. You have an array A copy it in another one B then sort array A in ascending order and the array B in descending order, then find the summation of the absolute difference between each element from A and the corresponding element from B.

O. You have an array 2D print the summation of the upper right triangle of the array.

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

P. You have an array 2D print the summation of the lower left triangle of the matrix.

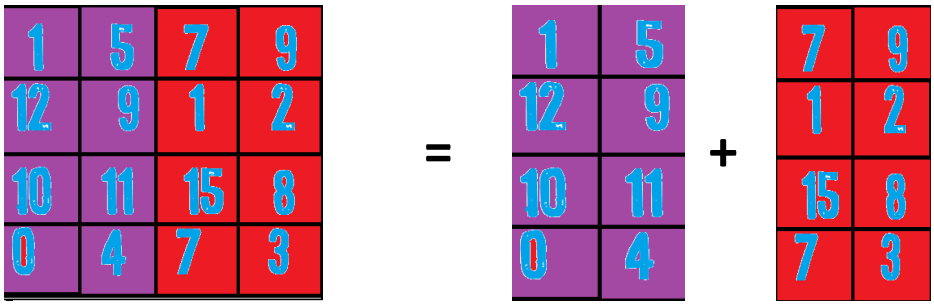
1	0	0	0
2	2	0	0
3	5	1	0
6	2	9	8

Q. You have a matrix and your task is to determine the absolute difference between the summation of corner elements and the center elements

(1+9+0+3)-(9+1+11+15) = 23

1	5	7	9
12	9	1	2
10	11	15	8
0	4	7	3

R. You have a matrix n*m print it after divide it int to arrays with equal dimensions :(assume that n and m is always even)



S. Can we swap an element of A with an element of B such that the sum of elements in the two arrays is equal with O (n) time complexity?

T. Implement selection sort

U. Implement bubble sort

24. What is the complexity of the following code??

A.

```
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
    int n;
    cin >> n;
    cout << n << endl;
     return 0;
 }
```

B.

```
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
   int n ;
     cin >>n;
   for (int i = 0; i < n; i++)
        cout << i << "\n";
     return 0;
 }
```

C.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
      int n ;
      cin >>n;
      for (int i = 0; i < n; i++) {
     for (int j = 0; j < n; j++)
          cout << j << " ";
     cout << "\n";
     }
      return 0;
 }
```

D.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
     int n;
     cin>>n;
     for (int i = 0; i < n; i++) {
          for (int j = 0; j < n; j++)
               cout << "Hello\n";
     }
     for (int i = 0; i < n; i++)
          cout << "Hello\n";
     return 0;
 }
```

E.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
     int n;
     cin>>n;
     for (int i = 0; i < n; i++) {
          for (int j = 0; j < n; j *= 2)
               cout << j << " ";
          cout << "\n";
     }
     return 0;
 }
```

F.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
     int n;
     cin>>n;
     for (int i = 1; i < n; i *= 2)
          cout << i << "\n";
     return 0;
 }
```

G.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
     int n;
     cin>>n;
     for (int i = 0; i < n; i++) {
          for (int j = 0; j < n; j += 2)
               cout << j << " ";
          cout << "\n";
     }
     return 0;
 }
```

H.

```cpp
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {   int n;
     cin>>n;
     for (int i = 0; i < n; i++){
          for (int j = 1; j < n; j *= 2)
               cout << j << " ";
          cout << "\n";
     }
     return 0;
 }
```

I.

```
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {  int i, j;
     for (i = 1; i <= n; i *= 2)
          for (j = 1; j <= i; j*=2)
               cout << "HEllO\n";
     return 0;
 }
```

J.

```
#include<iostream>
#include <cmath>
 using namespace std;
 int main()
 {
     for (int i = 0; i < n; i++)
          for (int j = 0; j < 1000; j++)
               cout << i * j;

     return 0;
 }
```

Answers

1.

a. Array declaration by specifying size

EX : int arr[10];

b. Array declaration by initializing elements

EX : int arr[] = { 10, 20, 30, 40 }

c. Array declaration by specifying size and initializing elements

EX : int arr[6] = { 10, 20, 30, 40 }

2. *(arr+5)
3. Yes
4. No , because it's static
5. If it's a dynamic array,We can change its size
6. Yes
7. arr[(n + 1) / 2 - 1]
8. Arr[0].
9. 5*4 (bytes)
10. 6*5*4 (bytes)
11. int arr [][][][][].
12. Square Brackets

13.

- Static Array :
  - ➢ Static arrays are allocated memory at compile time.
  - ➢ Size is fixed.
  - ➢ Located in stack memory space.
  - ➢ Eg. : int array[10]; //array of size 10

- Dynamic Array :
  - ➢ Memory is allocated at run time.
  - ➢ Size is not fixed.
  - ➢ Located in Heap memory space.
  - ➢ Eg. : int* array = new int[10];

14. Algorithmic complexity is concerned about how fast or slow a particular algorithm performs.

15. To express the time complexity of the worst-case scenario for a given algorithm.

16. To calculate the summation of numbers from 1 to n:
    1. Use a loop from 1 to n with complexity O (n)
    2. Use mathematical rule n*(n+1)/2 with complexity O(1)

17. Search by yourself

18. O(Log(n))

19. Sorted

20. Time complexity: O(n^2), Space complexity O(1)

21. Worst and Average Case Time Complexity: O(n*n)

Best Case Time Complexity: O(n)

22. What is the output?

A. The program may print 0 five times followed by garbage value, or may crash if address (arr+5) is invalid.
B. II. Compilation Error
C. Address of the first element in the array
D. run time error
E. 4
F. 10 ,Use this formula to add all numbers from 1 to n : n*(n+1)/2
G. random number
H. B. 6533.
I. 10 9 8 7 6 5 4 3 2 1
J. 5
K. 5 6 3 0 8 9 6 39
L. Random number then "ABC"
M. B. -30
N. Compilation Error
O. -1
P. Infinite loop

23. Problems :

The solutions is on GitHub

The solution is on GitHub

24. What is the complexity of the following??

A. O(1)
B. O(n)
C. O(n^2)

D. O(n^2)
   Explain :
   - The sum of steps is supposed to be (n + n ^ 2), but the Big-O notation is the element that has the largest impact on performance, which is the (n ^ 2)

E. O(oo), Explain: Infinity loop
F. O(log(n))
G. O(n^2)
H. O(nLog(n))
I. $O(\log(n)^2)$
J. O(n)