

Special Offer - All in One Software Development Bundle (600+ Courses, 50+ projects) @ **Shell sort C**

✕



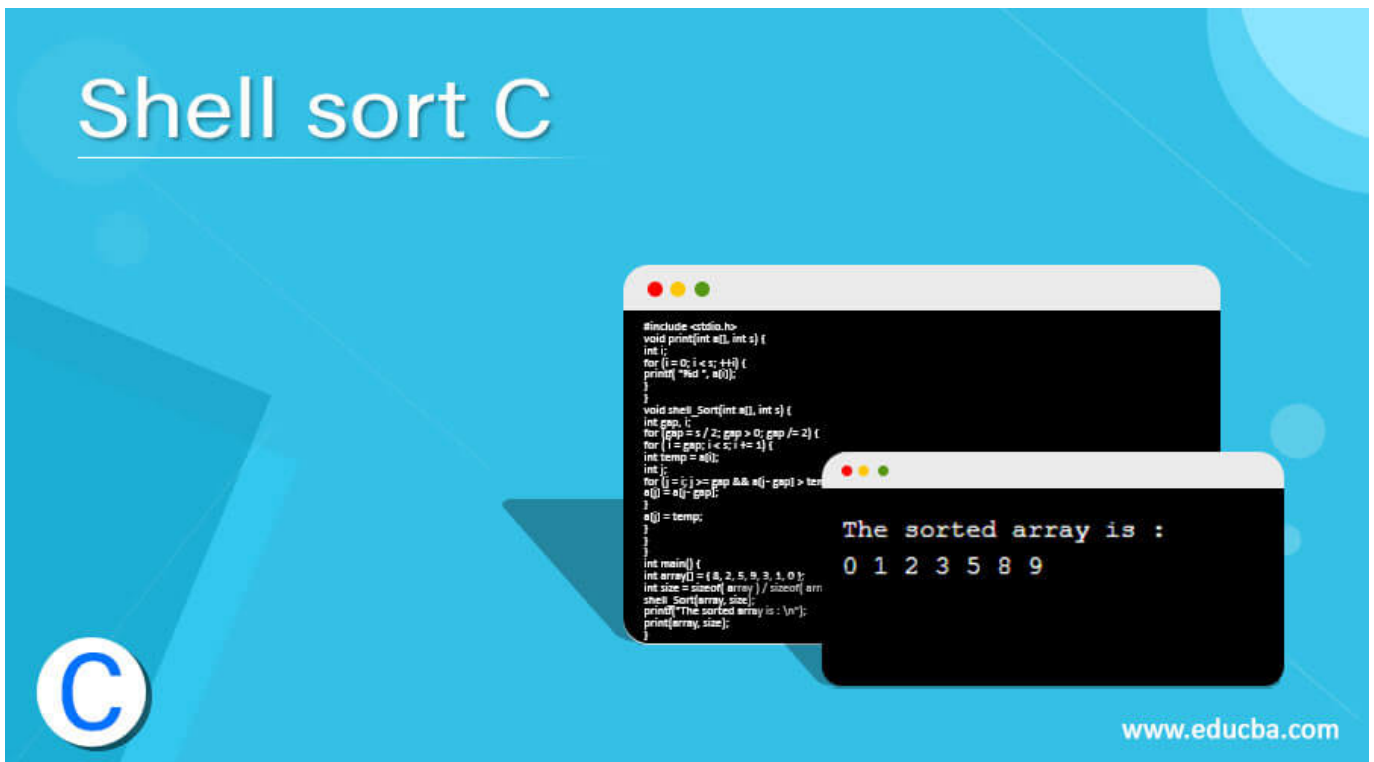
(<https://www.educba.com/software-development/>)



(<https://www.educba.com/selection-sort-in-c/>)



(<https://www.educba.com/constructor-in-c/>)



## Definition of Shell sort C

The Shell sort in C is used to sort the array by sorting the pair of elements far apart from each other, then successively reduces the gap between the elements to be sorted. The Shell sort is the version of the insertion sort algorithm. In insertion sort, the element moves one position



ahead to insert an element at its correct position, whereas the shell sort exchanges the far elements. If an element has to move far ahead many moves are required. The shell sort starts by sorting the pair of elements that are far apart from each other and successively interval <https://www.educba.com/software-development/> transfer some out-of-place elements into the correct place faster than a simple nearest-neighbor exchange if it starts with far apart elements. Shell sort is not a stable sorting algorithm since it ignores the items that fall in between the intervals. The worst-case running time complexity of the shell sort is  $O(n^2)$  and the best-case running time complexity is  $O(n \log(n))$ .

The algorithm of the shell sort –

### Start Your Free Software Development Course

Web development, programming languages, Software testing & others

```
Shell_Sort( a, length)
for interval x <- length / 2n down to 1
for each interval "x" in a
sort all the elements at interval "x"
end shell_Sort
```

Return value – The return value of this method is the sorted array.

## The working of the shell sort in C

working of the shell sort in C

1. Let the specified array is:

Given array: [8, 7, 2, 6, 4, 5, 4, 0]



2. In our algorithm, we use the shell's original sequence ( $N/2, N/4, \dots, 1$ ) as intervals. If the array

size  $N = 8$ , the elements in the interval of  $N/2 = 4$  are compared and swapped if they are out of order in the first loop.

The 0th element and the 4th element are compared.

(<https://www.educba.com/software-development/>)

If the 0th element is greater than the 4th, the 4th element is stored first in the temp variable, followed by the 0th element (i.e. greater element) in the 4th position, and the element stored in temp in the 0th position and rearrange the elements in an  $n/2$  interval.

array: [4, 7, 2, 6, 8, 5, 3, 0], temp = 4

This procedure is repeated for all the remaining elements and rearranges the elements in an  $n/2$  interval. we get the array:

🔗 Popular Course in this category



### C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,612 ratings)

Course Price

~~\$79~~ \$399

[View Course](https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

### Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)



**Special Offer - All in One Software Development Bundle (600+ Courses, 50+ projects) @ USD119**



array: [4, 5, 2, 0, 8, 7, 3, 6]

(<https://www.educba.com/software-development/>)

3. In second cycle, an interval of  $N/4 = 8/4 = 2$  is chosen, and the elements that fall within this range are sorted once more and rearrange the elements in an  $n/4$  interval.

array: [2, 5, 4, 0, 8, 7, 3, 6]

4. The 4th and 2nd place elements are compared. Also contrasted are the elements in the 2nd and 0th positions. The current interval is used to compare all of the elements in the sequence.

array: [2, 5, 4, 0, 8, 7, 3, 6]

This procedure is repeated for all the remaining elements and rearranges the elements in an  $n/2$  interval. we get the array:

array: [2, 0, 3, 5, 4, 7, 8, 6]

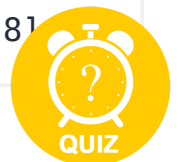
5. Next, the array elements in the interval of 1 are sorted when the interval is  $N/8 = 8/8 = 1$ . The array has now been sorted absolutely and rearrange the elements in an  $n/8$  interval.

array: [2, 0, 3, 5, 4, 7, 8, 6] array: [0, 2, 3, 5, 4, 7, 8, 6]

array: [0, 2, 3, 5, 4, 7, 8, 6] array: [0, 2, 3, 4, 5, 7, 8, 6]

array: [0, 2, 3, 4, 5, 7, 8, 6] array: [0, 2, 3, 4, 5, 7, 8, 6]

array: [0, 2, 3, 4, 5, 7, 8, 6] array: [0, 2, 3, 4, 5, 6, 7, 8]



## Examples for the shell sort in C

Example for shell sort in C to sort the array of numbers.

## Special Offer - All in One Software Development Bundle (600+ Courses, 50+ projects) @ USD119



```
#include <stdio.h>
void print(int a[], int s) {
    for (i = 0; i < s; ++i) {
        printf( "%d ", a[i]);
    }
}

void shell_Sort(int a[], int s) {
    int gap, i;
    for (gap = s / 2; gap > 0; gap /= 2) {
        for ( i = gap; i < s; i += 1) {
            int temp = a[i];
            int j;
            for (j = i; j >= gap && a[j - gap] > temp; j -= gap) {
                a[j] = a[j - gap];
            }
            a[j] = temp;
        }
    }

    int main() {
        int array[] = { 8, 2, 5, 9, 3, 1, 0 };
        int size = sizeof( array ) / sizeof( array[0] );
        shell_Sort(array, size);

        printf("The sorted array is : \n");
        print(array, size);
    }
}
```



**Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects)**



An output of the above code is -

```
The sorted array is :
0 1 2 3 5 8 9
```

[development/](https://www.educba.com/software-development/)

As in the above program, the shell\_Sort() function is created to sort the number array. Inside the function, the first for loop creates the half of the gap, the second loop perform a gapped insertion sort, if the first gap elements a[0..gap-1] are already gap sorted, then add one more element before the entire array is gap sorted, and the third for loop shift the earlier gap-sorted elements until the correct location for a[i] is found. Then finally, copy the temp to the original array and printing all the sorted array by using the print() function, as we can see in the above output.

Rewrite the above program to sort the character array.

## Example #2

```
#include <stdio.h>

void print(char a[], int s) {
    int i;
    for (i = 0; i < s; ++i) {
        printf( "%c ", a[i]);
    }
}

void shell_Sort(char a[], int s) {
    int gap,i;
    for (gap = s / 2; gap > 0; gap /= 2) {
        for ( i = gap; i < s; i += 1) {

            char temp = a[i];
            int j;
            for (j = i; j >= gap && a[j - gap] > temp; j -= gap) {
```



```

a[i] = a[j - gap];
}

```

```

a[j] = temp;
}

```

(<https://www.educba.com/software-development/>)

```

int main() {
    char array[] = { 'h', 'e', 'l', 'l', 'o' };
    int size = sizeof( array ) / sizeof( array[0] );
    shell_Sort(array, size);
    printf("The sorted array is : \n");
    print(array, size);
}

```

An output of the above code is –

```

The sorted array is :
e h l l o

```

As in the above program, the shell\_Sort() function is created to sort the character array. Inside the function, the first for loop creates the half of the gap, the second loop perform a gapped insertion sort, if the first gap elements a[0..gap-1] are already gap sorted, then add one more element before the entire array is gap sorted, and the third for loop shift the earlier gap-sorted elements until the correct location for a[i] is found. Then finally, copy the temp to the original array and printing all the sorted array by using the print() function, as we can see in the above output.

## Conclusion

The shell sort in C is used to sorts the array by sort the pair of elements far apart from each other, then successively reduces the gap between the elements to be sorted.



## Recommended Articles

## Recommended Articles

**Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects, 500+ Hours)** ×

This is a guide to Shell sort C. Here we discuss definition, syntax, and parameters, working of the shell sort in C examples with code implementation. You may also have a look at the

[following articles](https://www.educba.com/software-development/) to learn more –

1. [Sort String in Java](https://www.educba.com/software-development/) (<https://www.educba.com/sort-string-in-java/>)
2. [Sort string C++](https://www.educba.com/sort-string-c/) (<https://www.educba.com/sort-string-c/>)
3. [Bubble Sort in C](https://www.educba.com/bubble-sort-in-c/) (<https://www.educba.com/bubble-sort-in-c/>)
4. [Sort string PHP](https://www.educba.com/sort-string-php/) (<https://www.educba.com/sort-string-php/>)

## ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE (600+ COURSES, 50+ PROJECTS)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

**Learn More**

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>





**About Us**  
**Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects) @ USD119.00 only for 30 days** ×

 Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

<https://www.educba.com/software-development/>

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)

Contact Us (<https://www.educba.com/contact-us/?source=footer>)

Verifiable Certificate (<https://www.educba.com/software-development/verifiable-certificate/?source=footer>)

Reviews (<https://www.educba.com/software-development/reviews/?source=footer>)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

## Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

## Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)


Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)



**Certification Courses****Special Offer - All in One Software Development Bundle (600+ Courses, 50+**

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

 Software Development Course - All in One Bundle  
(<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)

Java Course (<https://www.educba.com/software-development/courses/java-course/?source=footer>)

Become a Selenium Automation Tester (<https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer>)

Become an IoT Developer (<https://www.educba.com/software-development/courses/iot-course/?source=footer>)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

