

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	35
J - h	0	81	i	5
	1	94	j	5
	2	11	h	5
	3	96		
	4	12		
i, j	5	35		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                → if (j < h)
                {
                    break;
                }
            }
            lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	35
J	0	35	i	5
	1	94	j	0
	2	11	h	5
	3	96		
	4	12		
i	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
	0	35	i	6
j-h	1	94	j	6
	2	11	h	5
	3	96		
	4	12		
	5	81		
i , j	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            aux = lista_k->item[i];
            j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                → if (j < h)
                {
                    break;
                }
            }
            lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
	0	35	i	6
j	1	94	j	1
	2	11	h	5
	3	96		
	4	12		
	5	81		
i	6	94		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            aux = lista_k->item[i];
            j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    → break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
	0	35	i	6
j	1	17	j	1
	2	11	h	5
	3	96		
	4	12		
	5	81		
i	6	94		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	95
	0	35	i	7
	1	94	j	7
j - h	2	11	h	5
	3	96		
	4	12		
	5	81		
	6	17		
i , j	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    → break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	28
	0	35	i	8
	1	94	j	8
	2	11	h	5
J - h	3	28		
	4	12		
	5	81		
	6	17		
	7	95		
i , j	8	96		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    → break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	28
	0	35	i	8
	1	94	j	3
	2	11	h	5
j	3	28		
	4	12		
	5	81		
	6	17		
	7	95		
i	8	96		
	9	58		
	10	41		
	11	75		
	12	15		



```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	58
	0	35	i	9
	1	94	j	9
	2	11	h	5
	3	96		
j - h	4	12		
	5	81		
	6	17		
	7	95		
	8	28		
i , j	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                → if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	41
	0	35	i	10
	1	94	j	10
	2	11	h	5
	3	96		
	4	12		
j - h	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
i , j	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	41
	0	35	i	10
	1	94	j	5
	2	11	h	5
	3	96		
	4	12		
j - h	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
i , j	10	81		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            aux = lista_k->item[i];
            j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	41
j - h	0	35	i	10
	1	94	j	5
	2	11	h	5
	3	96		
	4	12		
j	5	41		
	6	17		
	7	95		
	8	28		
	9	58		
i	10	81		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	75
	0	35	i	11
	1	94	j	11
	2	11	h	5
	3	96		
	4	12		
	5	81		
j - h	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
i, j	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	15
	0	35	i	12
	1	94	j	12
	2	11	h	5
	3	96		
	4	12		
	5	81		
	6	17		
j - h	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
i , j	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	15
	0	35	i	12
	1	94	j	12
	2	11	h	5
	3	96		
	4	12		
	5	81		
	6	17		
j - h	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
i , j	12	95		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            aux = lista_k->item[i];
            j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	15
	0	35	i	12
	1	94	j	7
J - h	2	11	h	5
	3	96		
	4	12		
	5	81		
	6	17		
j	7	15		
	8	28		
	9	58		
	10	41		
	11	75		
i	12	95		



```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	96
J - h	0	35	i	3
	1	94	j	3
	2	11	h	3
i , j	3	96		
	4	12		
	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {
    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	12
	0	35	i	4
J - h	1	94	j	4
	2	11	h	3
	3	96		
i , j	4	12		
	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                → if (j < h)
                {
                    → break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	12
	0	35	i	4
j	1	12	j	1
	2	11	h	3
	3	96		
i	4	94		
	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	81
	0	35	i	5
	1	94	j	5
J - h	2	11	h	3
	3	96		
	4	12		
i , j	5	81		
	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {

    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        → for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            while (lista_k->item[j-h].chave > aux.chave)
            {
                lista_k->item[j] = lista_k->item[j-h];
                j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
	0	35	i	6
	1	94	j	6
	2	11	h	3
j - h	3	96		
	4	12		
	5	81		
i , j	6	17		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {
    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
	0	35	i	6
	1	94	j	6
	2	11	h	3
j - h	3	96		
	4	12		
	5	81		
i , j	6	96		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {
    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
J - h	0	35	i	6
	1	94	j	3
	2	11	h	3
j	3	96		
	4	12		
	5	81		
i	6	96		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		

```

void shellSort (TypeLista *lista_k) {
    int i, j, h = 1;
    TypeItem aux;
    do
    {
        h = h * 3 + 1;
    } while (h < lista_k->ultimo);
    do
    {
        h = h/3;
        for (i = h; i < lista_k->ultimo; i++)
        {
            → aux = lista_k->item[i];
            → j = i;
            → while (lista_k->item[j-h].chave > aux.chave)
            {
                → lista_k->item[j] = lista_k->item[j-h];
                → j = j - h;
                if (j < h)
                {
                    → break;
                }
            }
            → lista_k->item[j] = aux;
        }
    } while (h != 1);
}

```

n = 13	Posição	Chave	AUX	17
j	0	17	i	6
	1	94	j	0
	2	11	h	3
	3	35		
	4	12		
	5	81		
i	6	96		
	7	95		
	8	28		
	9	58		
	10	41		
	11	75		
	12	15		