

```
1 #include <stdio.h>
2
3 int main() {
4
5     int emptyList[0];
6
7     int oneElementList[] = {42};
8
9     int identicalList[] = {5, 5, 5, 5, 5};
10
11    int negativeList[] = {-1, -2, -3, -4, -5};
12
13    printf("Empty List:\n");
14    for (int i = 0; i < sizeof(emptyList) / sizeof(emptyList[0]); i++) {
15        printf("%d ", emptyList[i]);
16    }
17
18    printf("\nOne Element List:\n");
19    for (int i = 0; i < sizeof(oneElementList) / sizeof(oneElementList[0]); i++)
20        ++
21    {
22        printf("%d ", oneElementList[i]);
23    }
24
25    printf("\nIdentical Element List:\n");
26    for (int i = 0; i < sizeof(identicalList) / sizeof(identicalList[0]); i++)
27        {
28            printf("%d ", identicalList[i]);
29        }
```

Empty List:  
One Element List:  
42  
Identical Element List:  
5 5 5 5 5  
Negative Number List:  
-1 -2 -3 -4 -5

Execute | Beautify | Share | Source Code | Help

```
1 #include <stdio.h>
2
3 void selectionSort(int arr[], int n) {
4     int i, j, minIndex, temp;
5     for (i = 0; i < n-1; i++) {
6         minIndex = i;
7         for (j = i+1; j < n; j++) {
8             if (arr[j] < arr[minIndex]) {
9                 minIndex = j;
10            }
11        }
12        temp = arr[i];
13        arr[i] = arr[minIndex];
14        arr[minIndex] = temp;
15    }
16 }
17
18 int main() {
19     int arr[] = {64, 25, 12, 22, 11};
20     int n = sizeof(arr)/sizeof(arr[0]);
21     selectionSort(arr, n);
22     printf("Sorted array: \n");
23     for (int i=0; i < n; i++) {
24         printf("%d ", arr[i]);
25     }
26     return 0;
27 }
28
```

Terminal

Sorted array:

11 12 22 25 64

```
1 #include <stdio.h>
2
3
4 void bubbleSort(int arr[], int n) {
5     for (int i = 0; i < n-1; i++) {
6         for (int j = 0; j < n-i-1; j++) {
7             if (arr[j] > arr[j+1]) {
8                 int temp = arr[j];
9                 arr[j] = arr[j+1];
10                arr[j+1] = temp;
11            }
12        }
13    }
14 }
15
16 int main() {
17     int arr[] = {64, 34, 25, 12, 22, 11, 90};
18     int n = sizeof(arr)/sizeof(arr[0]);
19     bubbleSort(arr, n);
20     printf("Sorted array: ");
21     for (int i = 0; i < n; i++)
22         printf("%d ", arr[i]);
23     return 0;
24 }
25
```

```
Sorted array: 11 12 22 25 34 64 90
```

```
1 #include <stdio.h>
2
3 void insertionSort(int arr[], int n) {
4     int i, key, j;
5     for (i = 1; i < n; i++) {
6         key = arr[i];
7         j = i - 1;
8
9         while (j >= 0 && arr[j] > key) {
10             arr[j + 1] = arr[j];
11             j = j - 1;
12         }
13         arr[j + 1] = key;
14     }
15 }
16
17 int main() {
18     int arr[] = {12, 11, 13, 5, 6};
19     int n = sizeof(arr) / sizeof(arr[0]);
20
21     insertionSort(arr, n);
22
23     printf("Sorted array: \n");
24     for (int i = 0; i < n; i++) {
25         printf("%d ", arr[i]);
26     }
27     return 0;
28 }
```

Sorted array:  
5 6 11 12 13

Execute | Beautify | Share | Source Code | Help

```
1 int findKthPositive(int* arr, int arrSize, int k) {
2     int missingCount = 0;
3     int current = 1;
4     int i = 0;
5
6     while (missingCount < k) {
7         if (i < arrSize && arr[i] == current) {
8             i++;
9         } else {
10            missingCount++;
11        }
12        if (missingCount == k) {
13            return current;
14        }
15        current++;
16    }
17
18    return -1; // If k exceeds the missing count
19 }
20
```

Terminal

```
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/Scrt1.o: in
      function '_start':
(.text+0x1b): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

Execute | Beautify | Share | Source Code | Help

```
1 int findPeakElement(int* nums, int numsSize) {
2     int left = 0, right = numsSize - 1;
3
4     while (left < right) {
5         int mid = left + (right - left) / 2;
6
7         if (nums[mid] < nums[mid + 1]) {
8             left = mid + 1;
9         } else {
10            right = mid;
11        }
12    }
13
14    return left;
15 }
16
17
```

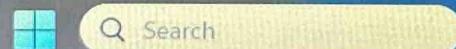
Terminal

```
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linu
      function `__start':
(.text+0x1b): undefined reference to `main'
collect2: error: ld returned 1 exit status
```

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int findNeedleInHaystack(char* haystack, char* needle) {
5     char* result = strstr(haystack, needle);
6     if (result == NULL) {
7         return -1;
8     } else {
9         return result - haystack;
10    }
11 }
12
13 int main() {
14     char haystack[] = "Hello, this is a haystack example.";
15     char needle[] = "haystack";
16
17     int index = findNeedleInHaystack(haystack, needle);
18     printf("Index of the first occurrence: %d\n", index);
19
20     return 0;
21 }
```

javascript:void(0)

Index of the first occurrence: 17



78°F  
Mostly cloudy

Execute | Beautify | Share | Source Code | Help

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int isSubstring(char *str, char *substr) {
5     return strstr(str, substr) != NULL;
6 }
7
8 void findSubstrings(char **words, int size) {
9     for (int i = 0; i < size; i++) {
10         for (int j = 0; j < size; j++) {
11             if (i != j && isSubstring(words[j], words[i])) {
12                 printf("%s is a substring of %s\n", words[i], words[j]);
13             }
14         }
15     }
16 }
17
18 int main() {
19     char *words[] = {"apple", "banana", "lemon", "on", "an"};
20     int size = sizeof(words) / sizeof(words[0]);
21     findSubstrings(words, size);
22     return 0;
23 }
24
25
26
27
```

javascript:void(0)

Terminal

on is a substring of lemon  
an is a substring of banana