1.Horspool's Algorithm

```
#include <stdio.h>
#include <string.h>

#define MAX 256

void shiftTable(char pattern[], int shift[]) {
    int m = strlen(pattern);
    for (int i = 0; i < MAX; i++) {
        shift[i] = m;
    }
    for (int j = 0; j < m - 1; j++) {
        shift[(int)pattern[j]] = m - 1 - j;
    }
}

int horspool(char text[], char pattern[]) {
    int shift[MAX];
    int n = strlen(text);
    int m = strlen(pattern);

    shiftTable(pattern, shift);

    int i = m - 1;
    while (i < n) {
        int k = 0;
```

```
        while (k < m && pattern[m - 1 - k] == text[i - k]) {
            k++;
        }
        if (k == m) {
            return i - m + 1; // Pattern found at index i - m + 1
        }
        i += shift[(int)text[i]];
    }
    return -1; // Pattern not found
}

int main() {
    char text[100], pattern[50];

    printf("Enter the text: ");
    gets(text);
    printf("Enter the pattern: ");
    gets(pattern);

    int result = horspool(text, pattern);
    if (result != -1) {
        printf("Pattern found at index %d\n", result);
    } else {
        printf("Pattern not found\n");
    }
```

```
    return 0;
}
```

Output:

Enter the text: this is a simple example

Enter the pattern: example

Pattern found at index 17

2.Heap sort

```
#include <stdio.h>

// Function to swap two elements
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Function to heapify a subtree rooted with node i
void heapify(int arr[], int n, int i) {
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;

    if (left < n && arr[left] > arr[largest]) {
        largest = left;
    }
```

```
    if (right < n && arr[right] > arr[largest]) {
        largest = right;
    }

    if (largest != i) {
        swap(&arr[i], &arr[largest]);
        heapify(arr, n, largest);
    }
}

// Main function to do heap sort
void heapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--) {
        heapify(arr, n, i);
    }

    for (int i = n - 1; i >= 0; i--) {
        swap(&arr[0], &arr[i]);
        heapify(arr, i, 0);
    }
}

void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
```

```
    printf("\n");
}

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    heapSort(arr, n);

    printf("Sorted array is:\n");
    printArray(arr, n);
    return 0;
}
```

Output:

Enter the number of elements: 5

Enter the elements of the array:

4 10 3 5 1

Sorted array is:

1 3 4 5 10